Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**
4. TITLE AND SUBTITLE **A User's Guide to the Technical Training Schoolhouse Model**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Rand Corporation,1776 Main Street,PO Box 2138,Santa Monica,CA,90407-2138**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**99**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

This product is part of the RAND Corporation technical report series. Reports may include research findings on a specific topic that is limited in scope; present discussions of the methodology employed in research; provide literature reviews, survey instruments, modeling exercises, guidelines for practitioners and research professionals, and supporting documentation; or deliver preliminary findings. All RAND reports undergo rigorous peer review to ensure that they meet high standards for research quality and objectivity.

TECHNICAL
REPORT

A User's Guide to the Technical Training Schoolhouse Model

Thomas Manacapilli, Bart Bennett

Prepared for the United States Air Force

Approved for public release; distribution unlimited

RAND PROJECT AIR FORCE

The research reported here was sponsored by the United States Air Force under Contracts F49642-01-C-0003 and FA7014-06-C-0001. Further information may be obtained from the Strategic Planning Division, Directorate of Plans, Hq USAF.

Library of Congress Cataloging-in-Publication Data

Manacapilli, Thomas.
A user's guide to the technical training Schoolhouse Model / Thomas Manacapilli, Bart Bennett.
p. cm.
Includes bibliographical references.
ISBN 978-0-8330-3997-2 (pbk. : alk. paper)
1. Aeronautics, Military—Study and teaching—United States—Simulation methods. I. Bennett, Bart, 1959– II. United States. Air Force. Air Education and Training Command. III. Project Air Force (U.S.) IV. Rand Corporation. V. Title.

UG638.M32 2006
358.4'150973—dc22

2006036764

Published 2006 by the RAND Corporation
1776 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138
1200 South Hayes Street, Arlington, VA 22202-5050
4570 Fifth Avenue, Suite 600, Pittsburgh, PA 15213-2665
RAND URL: http://www.rand.org/
To order RAND documents or to obtain additional information, contact
Distribution Services: Telephone: (310) 451-7002;
Fax: (310) 451-6915; Email: order@rand.org

Preface

In 2002, RAND Project AIR FORCE studied the data systems used by the Air Education and Training Command (AETC) to manage training costs and capacities. One recommendation from the study concluded that AETC lacks analytical tools to evaluate changes to the technical training pipeline. The schoolhouse model grew out of this recommendation. The model specifically examines resources used and training limitations encountered during the execution of a training program. The schoolhouse model, along with other training-oriented tools developed at the RAND Corporation, is intended to investigate the policy implications of numerous technical training pipeline issues.

At the same time, the AETC Studies and Analysis Squadron (SAS) built a similar set of planning and execution assessment tools in the context of a larger suite of models to develop Program Objective Memorandum costs. RAND and AETC SAS mutually agreed to combine the schoolhouse portion of their efforts into one model for both organizations. AETC SAS continued developing a variety of other models with the RAND schoolhouse model as a central core of its suite.

The purpose of this report is to provide users of the schoolhouse model with a reference for collecting and implementing data in the Microsoft® Excel® front end. This report also briefly describes the Extend™ simulation model.[1] The principal audience for this report is the analysts who are studying issues related to training pipeline resource requirements. Familiarity with Microsoft Excel is required.

Prior RAND Project AIR FORCE research on AETC training systems was published as *Air Education and Training Command Cost and Capacity System: Implications for Organizational and Data Flow Changes* by Thomas Manacapilli et al. (MR-1797-AF). That report develops a four-level model of management to evaluate the flow of data in the AETC training pipeline. The resulting conclusions include a recommendation to consolidate strategic management functions to resolve data flow problems, and the use of methodological tools, such as simulations, to evaluate trade-offs in the training pipeline.

The research reported here was sponsored by the Air Force Deputy Chief of Staff for Personnel (AF/DP) and the Commander, Air Education and Training Command (AETC/CC)

[1] Extend is a trademark of Imagine That, Inc.

and conducted within the Manpower, Personnel, and Training Program of RAND Project AIR FORCE. It was part of a fiscal year 2005 project, Cost and Productivity of Technical Training Versus On-the-Job Training Analysis.

RAND Project AIR FORCE

RAND Project AIR FORCE (PAF), a division of the RAND Corporation, is the U.S. Air Force's federally funded research and development center for studies and analyses. PAF provides the Air Force with independent analyses of policy alternatives affecting the development, employment, combat readiness, and support of current and future aerospace forces. Research is conducted in four programs: Aerospace Force Development; Manpower, Personnel, and Training; Resource Management; and Strategy and Doctrine.

Additional information about PAF is available on our Web site at http://www.rand.org/paf.

Contents

CHAPTER THREE

CHAPTER FOUR

Figures

Tables

Summary

Prior RAND research on AETC's management of enlisted training identified a lack of analytic tools to assess changes in the pipeline (Manacapilli et al., 2004). That study recommended the development of two models. First, AETC needs a model of an actual schoolhouse system. Second, AETC needs a model of the entire training pipeline, from recruiting to on-the-job training at the operational units. The schoolhouse model grew out of the first recommendation. This document is a user's guide for the Excel-based front end used to organize and manage the detailed data for the Extend simulation model. (See pp. 1–3.)

AETC manages initial skills training across the Air Force. Broadly speaking, this consists of flying training and technical training across a wide range of career fields. Compared with flying training, technical training is a small training component in terms of dollars spent but huge in terms of the number of people trained. (See pp. 5–7.) Flying training can cost \$1 million or more per pilot, but only slightly more than 1,000 pilots per year undergo flying training. Technical training averages just \$20,000 per student, but more than 30,000 students receive technical training in initial skills alone. Consequently, understanding and improving the operation of the technical training pipeline can have a significant impact on Air Force costs and on the quality of airmen undertaking their first assignments. (See p. 8.)

We developed the schoolhouse model to assist in the planning and resourcing technical training. The model provides an entity-level simulation of an actual training group and its associated squadrons. The model simulates courses, plans of instruction, flights, instructors, training devices, and classroom facilities. Analysts can use the schoolhouse model to develop estimates of the resource requirements for initial skills training courses. Its uses include

- evaluation of the change in production with increases or decreases in resources (facilities, instructors, and training devices)
- highlighting resource bottlenecks as a result of changes in the plan of instruction
- providing insight into classroom details such as the ratio of empty seats to the average number of individuals who prove ineffective in training
- assessing the change in production resulting from changes in washback and attrition rates. (See pp. 9–12.)

These are only a handful of the many ways in which an analyst can use the model to evaluate initial skill training issues.

The schoolhouse model is composed of two applications. The front end of the model contains a set of Excel worksheets along with Microsoft Visual Basic® programs that control the input and manipulation of the data. The second application is an Extend simulation model of the schoolhouse processes. (See pp. 13–17.)

The model has been purposefully designed to be data-driven. This means that neither the Visual Basic routines nor the Extend model must be rewritten in order to analyze a different context or a different schoolhouse. The data fully define the structure and operation of the schoolhouse. (See pp. 19–46.)

The Excel front end and the Extend model mimic current AETC processes. For example, we use data formats taken directly from AETC databases, manuals, and forms. (See pp. 9–12.) The data are represented in a way that is very similar to actual forms and data formats used by AETC. Additionally, the model follows the processes defined for technical training.

We chose Excel as the front end because of its widespread use and extensive capabilities. It offers a user-friendly interface that can handle a wide variety of data types in one file location. Additionally, the embedded Visual Basic capability allowed us to build routines to convert the data into text files for use in the Extend simulation.

The development of the schoolhouse model has three main strengths over previous methods. First, AETC has little capability to model the technical training process and so any repeatable mathematical tool is a marked increase over the present capability. Second, AETC manages hundreds of different courses. It is not feasible to build a model for every course. As mentioned previously, the schoolhouse model is data-driven: No new coding is required to model a different course. (See pp. 51–52.) Finally, the model produces a detailed history file of every event in the simulation. (See pp. 20, 48–49.) The model need not be rerun to look at other measures or metrics. Instead, the event history file can be reanalyzed with statistical tools. (See pp. 53, 58–59.)

The weakness of the schoolhouse model falls into three areas. (See pp. 51–52.) First, it is time-consuming to build the databases. It can take from one day to one week to gather and input all the data required for a course. The data required to run the model are readily available, although it must be obtained from multiple sources and translated from multiple formats. Future enhancements to the model may include an automated data-building feature. Already, AETC SAS has developed some automated tools to build parts of the database.

The second major weakness is the long run time for analysis. An analysis of changes at a typical wing can require run times of 10 to 20 computer hours. Dual-processor computers or multiple computers can reduce the time required.

The final weakness is the very large size of the event history file. As noted above, this file is extremely useful for analysis. Unfortunately, the file is very large. One two-hour run can easily produce a 100- to 200-MB file. Multiple replications require gigabytes of storage. The entire AETC pipeline may require terabytes of storage.

The schoolhouse model has many applications. It is currently a working model, but it can be enhanced to include additional tools and training options. A next step is to create a user's group to guide the development and future use of the model. (See pp. 61–62.)

Acknowledgments

Our greatest appreciation is due to Mike Snedeker (AETC A5/8A) who had the original vision of improving AETC planning processes. Lt Col Jim Boyd (AETC A5RI) did a lot of the legwork to satisfy administrative requirements and make our job easier.

We wish to especially thank the staff of AETC SAS. The squadron provided an incredible amount of energy, enthusiasm, and constructive criticism. Our special thanks to Lt Col Lee Williams, Lt Col Bonnie Lind, Capt Chris Chocolaad, Capt June Rodriguez, Capt Susan Storm, Capt Zabrina Hoggard, Capt Paul Clemans, Lt Tanya Garcia, Lt Erika Banks, and SSgt Leonard Burkhardt.

Additionally, the staff at the 37th TRG provided extensive support. Our special thanks is given to TSgt Rhonda Battles, Gene Hoffman, Beth Robinson, Sam Zamarripa, and MSgt Susan Koutrouba.

Finally, we wish to thank our reviewers, Pat Bowden and Manuel Carrillo, for their extremely helpful comments, criticisms, and insights. Their input made the report a much more useful product.

Abbreviations

AETC	Air Education and Training Command
AETC/CC	Commander, Air Education and Training Command
AF/DP	Air Force Deputy Chief of Staff for Personnel
AFSC	Air Force specialty code
BIC	basic instructor course
BMT	basic military training
IIT	ineffective in training
IST	initial skills training
MCTR	mean cost to repair
MTBF	mean time between failures
MTTR	mean time to repair
MUBB	mean uses between breaks
PAF	Project AIR FORCE
PCD	program control data
PDS	personnel data system
PGL	program guidance letter
PM	program manager
POI	plan of instruction
POM	program objective memorandum
PPBS	planning, programming, and budgeting system
SAS	Studies and Analysis Squadron
SAT	student awaiting training

TM	training manager
TPR	trained personnel requirements
TPS	Training Planning System
TRG	training group
TRQI	training resource quantity indicator
TRS	training squadron
UOI	unit of instruction

CHAPTER ONE

Introduction

The purpose of this report is to provide a user's guide for organizing and managing the detailed training data necessary to assess policy options for resourcing the training pipeline. A front-end interface for the model using Excel provides a convenient means of manipulating the input data in preparation for the schoolhouse simulation model. We also briefly describe the Extend-based simulation model and model outputs. This report serves as a reference manual for analysts familiar with quantitative analysis.

This user's guide will be most useful to analysts with knowledge of the Air Force technical training process, a background in discrete simulation, and knowledge of statistical tools for evaluating the output. They will be able to use this model to provide resource planners with better insight into technical training pipeline issues, including key measures of throughput and cost.

Background

In 2002, the RAND Corporation evaluated the data systems supporting cost and capacity assessments for the Air Education and Training Command (AETC) (Manacapilli et al., 2004). The study concluded that AETC lacked analytical tools to evaluate changes to the technical training pipeline and recommended the development of new tools. RAND developed the schoolhouse model in response to that recommendation.

Objectives and Approach

The schoolhouse model has been designed to analyze many aspects in determining technical training resource requirements, including the sharing of resources among training units, utilization of resources, resource limitations, and resource constraints. The model explicitly represents instructors, classrooms, facilities, and training devices as the primary resources. The model can predict changes in production due to syllabus or resource changes. It can be used to analyze the impact of changes in washouts and washbacks on seat set-asides.[1]

[1] A *washout* is a student who does not complete the prescribed course. In some cases, the individual is reclassified into another Air Force specialty code (AFSC); in other cases, the individual is discharged from the Air Force. A *washback* is

The schoolhouse model focuses on the technical training of airmen. It does not model the recruitment process, basic military training, or on-the-job training. While the model is applicable to officer technical training, the original purpose was the training of enlisted airmen. The model was not designed for flying training.

The model uses two commercial applications. The front-end (data input) portion uses Microsoft Excel.[2] The model features specific Visual Basic routines to exploit the graphical nature of Excel to simplify the data input process. Additionally, the front end exports the data into an intermediary form for input into the simulation engine. The simulation engine is built using Extend, a commercial simulation language. We modified the software of standard Extend blocks to develop a simulation very specific to a military training environment. The model utilizes intermediary text files to transfer data from the Excel front end to Extend and the output data from Extend.

The methodology for the simulation model is straightforward. The model simulates the flow of flights (groups of students) through the course plan of instruction (POI), including the requisite amount of resources (instructors, facilities, training devices, and time) according to the course POI. The model produces an event history from which we can calculate facility, instructor, and device usage rates. We can compute such aggregate statistics as the graduate production and investigate details such as bottlenecks in the POI.

Building a *data-driven* model was the key philosophical approach in developing the schoolhouse model. Scenario and policy changes can be evaluated by changing the model's input data. Once the essential data are in the right format, the schoolhouse model can be used to simulate any Air Force schoolhouse.

In contrast, the AETC Studies and Analysis Squadron (SAS) built a simulation of the navigator schoolhouse in early 2000.[3] The model had hundreds of nodes representing each piece of instruction. If the syllabus (POI for aircrews) changed, the model required significant reprogramming in order to implement the change. To model each AETC course in this way would require hundreds of individual models and a large group of dedicated computer programmers to make updates to the models as the courses changed. We specifically designed the RAND schoolhouse model to avoid this pitfall.

Organization of This Report

Chapter Two provides an overview of a training schoolhouse and our concept for representing the schoolhouse in a model. We also describe how the model can be used in the planning pro-

an individual who has failed some block of instruction in the course. The individual "washes back" to the next flight with an open seat, reentering his or her failed block of instruction. If no seat exists, the individual remains in IIT (ineffective-in-training) status. There are a number of programs to keep IITs busy until an opportunity to restart the block occurs.

2 There is a multitude of good software tools and packages that we could have used to build the front-end graphical interface. We chose Microsoft Excel due to its widespread use, availability, and familiarity.

3 One of the authors was commander of AETC SAS at the time and commissioned the navigator simulation development.

cess and provide a general discussion of the required data. Chapter Three describes the Excel front end, a tool for inputting all the required data. Chapter Four includes a discussion of the Extend simulation model at a very macro level. Chapter Five provides a summary of the current uses and current and potential applications of the model.

Note for the First-Time User

To use the model, there are some very specific requirements, explained in Chapter Three. At a very basic level, the user requires Excel and at least a player version of Extend.[4] Excel is included in the standard Microsoft Office package. The Extend player version, available from Imagine That, Inc., allows users to run Extend models but does not allow changes to the model.

[4] Technical requirements are outlined in Appendix A.

CHAPTER TWO

The Schoolhouse Model System

Overview of a Technical Training Schoolhouse

A technical training schoolhouse more closely resembles a factory than a university. Whereas in a university individuals select courses to satisfy their educational needs, a technical training schoolhouse groups students into flights that take a prescribed set of classes in order to reach an initial qualification in a particular specialty. There is no individuality within a specialty. Therefore, as in a factory production line, an object goes through a set of specific processes and emerges as a finished product. Figure 2.1 provides a concise overview of the process as it is framed in the RAND schoolhouse model.

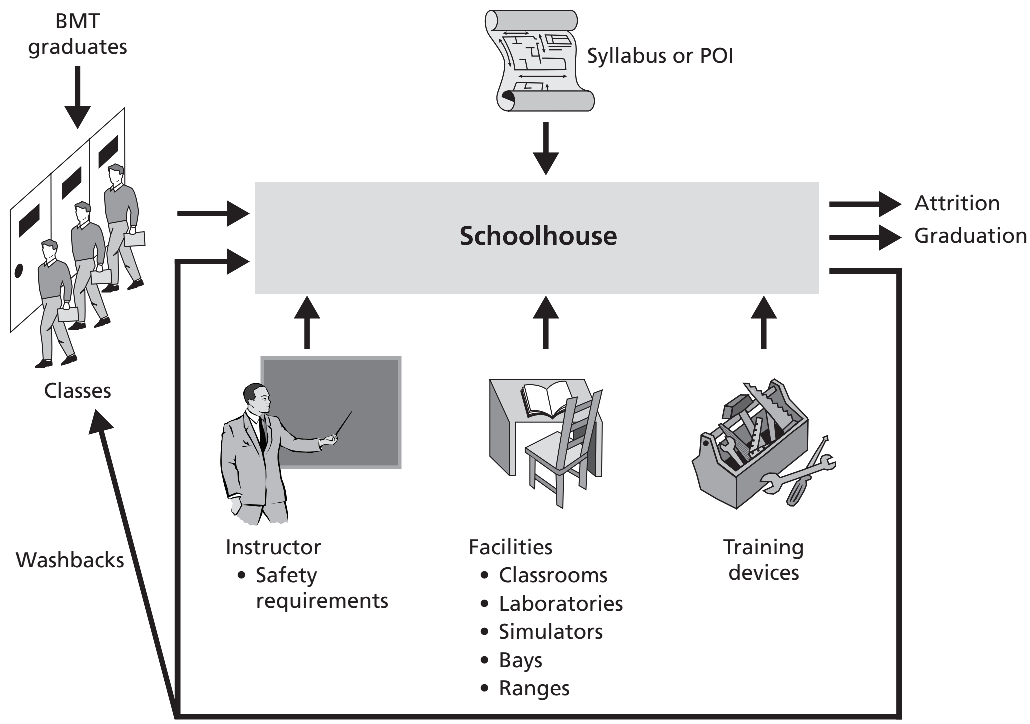
Students graduate from basic military training (BMT), where they receive broad, fundamental skills and are classified by Air Force specialty. The graduates are then transported to the Air Force bases that provide the necessary initial skills training (IST) for their respective specialties. In the majority of cases, trainees are preselected for their specialties as part of their recruitment. Thus, the Air Force knows their specialties prior to sending them to BMT and can schedule their entrance into BMT such that their graduation from BMT will occur immediately prior to the start of their first IST class. In cases in which classification does not match up with a class start date, trainees will travel to their respective IST bases and await an available opening in the first class in the sequence of their training courses. The Air Force keeps these trainees busy with a variety of activities and tasks, but it is desirable to reduce this waiting time as much as possible. In either case, significant resources begin to be allocated to these trainees as soon as they reach the base for such necessities as dorms for housing and the use of dining facilities.

The Air Force groups students attending the same specialty training into flights. The flights are part of a larger squadron, also called a schoolhouse.[1] For the most part, the flight remains together—eating, attending classes, marching, and graduating together. The Air Force houses the flight members in close proximity to one another—for example, setting aside a whole floor of a dorm for the flight.[2] The Air Force places a great deal of importance on unit

[1] The Air Force uses the word *squadron*. This report uses *squadron* interchangeably with *schoolhouse*.

[2] Male and female living arrangements are kept separate.

Figure 2.1
Overview of the RAND Schoolhouse Model



RAND *TR378-2.1*

integrity at the expense of some efficiency. If dorm space is short, airmen will triple-bunk in rooms.[3] The Air Force will also stagger eating times to maximize dining hall usage.

The flight's schedule of instructional topics is defined by the POI. The POI is often broken into large pieces of thematic content, called *blocks*, and then into smaller subsets that the Air Force calls *course content*. For each unit of course content, the POI defines the length of the various aspects of instruction (e.g., demonstration, lecture, application), the total length of the instruction, the required number and type of training devices, the required number and type of facilities, and the number of instructors. Often, a piece of the course content will have extra instructors for safety reasons. These are all defined in the POI.

The POI also spells out the evaluation events. Failure in an evaluation event, such as a written test, may result in a washback. The student is taken out of the flight and put into IIT status. The student will remain IIT, awaiting another flight with an available space and entering the same block of training from which the student failed. If an open seat exists, the student will join the new flight and become part of that flight. A certain number of seats is generally set aside for washbacks. Too many seats set aside for washbacks reduces production; not enough seats results in a longer wait time before students can return to training. A student

---

3 The Air Force training standard is a double-bunk arrangement.

can wash back multiple times depending on the policies in the schoolhouse and the judgment of schoolhouse leadership. Washback rates are fairly consistent and usually are tied to an evaluation event.

A student can also wash out and be eliminated from the training program in that specialty. While washouts can occur because of repeated course failure or an inability to grasp the material, most washouts are due to discipline and medical issues. Consequently, a washout can occur at any point in the course. Depending on the reason for the washout, it is possible to reclassify a washout into another specialty. In most cases, repeat washouts are given a general discharge from the Air Force. Washouts also open up seats for potential washbacks, except in the last block of instruction.[4]

Generally, flights follow the POI verbatim, but sometimes special conditions arise that justify a deviation. For example, if a critical training device breaks, it is possible to reorder course content within a given block. When a training device or certain facilities are not available, two other options are also possible. If the item is critical for skill training, the schoolhouse can assign a "training deficiency" to the student training record. The student would then receive the missing training at his or her next base. If the item is not critical, the schoolhouse can work around the missing item, and the student would still receive credit for the instruction.

The Air Force, in general, does not delay technical training because of missing equipment or facilities. The philosophy is to meet start and end dates for training, to handle resource limitations and exceptions as well as is possible within these dates, and to deal with more extreme problems outside the schedule.

Upon successful completion of all the items in the POI, the flight graduates, freeing up dorm space and reducing dining room usage. The students are awarded a 3-level classification and sent to their new bases where they continue their training on the job.

Sufficiently resourcing technical training is the primary objective of our analyses. In representing the schoolhouse, we focus on the primary resources: instructors, facilities, and training devices. In many cases, these resources are shared only among the various classes for a given course. However, some resources, such as instructors, the pool, the gym, and the firing range may be shared among multiple courses for the same specialty or across specialties at the same base. Providing too many of these shared resources would be a waste of valuable training funds. Provide too few and quality of training would be severely degraded. The schoolhouse model was specifically designed to address this balance. Additionally, the schoolhouse model must fit within the larger context of strategic training decisionmaking.

Overview of Processes Directly Affecting the Schoolhouse

Two primary strategic planning processes affect the training process. The first determines the requisite number of people to train in each Air Force specialty each year. The second defines the training budget. Each of these processes is discussed below.

---

4 If an individual washes out of the last block, this does not open up a seat, since the flight will have already started the block or will soon be graduating, so there is no opportunity to fill the available seat.

Trained Personnel Requirements Process

The purpose of the trained personnel requirements (TPR) process is to define the training requirements of each schoolhouse in terms of the number of BMT graduates entering IST. The TPR process produces the program guidance letter (PGL). The PGL represents the mission or requirement of the schoolhouse. The TPR process is long, beginning more than a year before students arrive at IST. The process starts at the Pentagon and uses computer models that consider staffing, authorization changes, attrition, reenlistment rates, current pipeline projections, projected crossflows, year group sizes, priorities, requirements from other services and government organizations, and other factors to produce an initial trained personnel requirement. AETC then evaluates the initial TPR against capacity constraints, construction projects, training device maintenance issues, and other issues during an AETC-wide TPR conference. The Air Staff takes the results of the conference and publishes the PGL.

AETC takes the PGL and begins developing a course schedule that will meet the requirements defined in it. Recruiting takes the course schedule and develops a plan for accessing individuals into the Air Force in time to graduate from BMT and then attend IST.

Planning, Programming, and Budgeting System Process

The TPR process is really a subset of the larger planning, programming, and budgeting system (PPBS) process. The unified and specified commanders assess the potential threats against the guidance given by the President to develop plans and against the corresponding capabilities and resources needed to execute those plans. The services integrate these operational requirements with projected fiscal, personnel, and material resources. There are never enough resources to meet all the operational requirements, so the services accept some level of risk in their fiscal plans.

The fiscal plans then define the personnel needs and training requirements of the services. Changes in the economy, retention, and many other factors make the personnel side of PPBS a dynamic process. The Air Force needs accurate estimates of training production capacity, and the costs of training need to properly balance the risks, required resources, and the needs of the Air Force as a whole.

Overview of Potential Output

The tool described in this user's guide has the potential to assist the planner in several planning processes. The schoolhouse model can predict resource utilization, instructor requirements, facility utilization and requirements, training device utilization and requirements, skill production, and numerous cost measures (total, training devices, facility, attrition, washback, IIT, and student-awaiting-training [SAT] costs). All these quantities play a role in determining the cost, quality, and timeliness of trainees. Because the schoolhouse model looks both within and across training squadrons at a base, the most efficient and effective set of resources can be determined. Planners can use the schoolhouse model to examine the cost and resource implications of changes for specialties or courses taught at a base, increases in course lengths, inclusion or exclusion of specific skills in a POI, and the imposition of higher standards.

The Excel Front End

The Excel front end provides a convenient interface to collect, browse, and manipulate the large quantity of data necessary to capture the salient characteristics of IST. Sheets are used to organize the data according to broad data groupings. Excel provides a visual layout that is easy to read and use. The front end then creates text input files for use by the simulation engine. The Excel front end replicates a number of familiar AETC forms to make the data input task easier. The front end allows for the quick creation of new scenarios by altering a set of baseline data. It also builds a logical storage system for scenarios that provides a historical audit trail of previous cases.

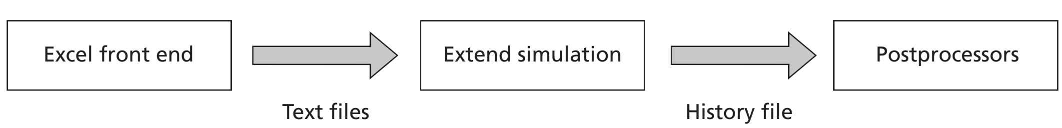
The Extend Simulation

The Extend simulation is intended to be a "behind-the-curtain" tool that most users will never need to modify or view. The schoolhouse model is data-driven, and all changes to the model are accomplished in the Excel front end. Figure 2.2 visually portrays the flow of data between the main parts of the schoolhouse model. Data are entered in the Excel front end and then passed to the simulation engine through intermediary text files. The simulation outputs a history file of the events for postprocessing and stores the results in the scenario directory, thereby keeping related input and output data in the same place. A more detailed description of the simulation is found in Chapter Four.

Data Requirements

The schoolhouse model requires data describing the resources, syllabi, and operations of the training courses and organizations. These data are gathered from a variety of sources. The Excel data input sheets are designed to replicate similar Air Force forms already used to record the specific data items, when applicable. We chose these formats because most of the data are found in these Air Force products, and we wanted to make the data entry as familiar and straightforward as possible. We have divided the data requirements of the model into six areas: group-level, facility, instructor, squadron-level, course plans, and training devices. The following sections describe the data in general terms and the type of policy questions or analyses that can result from changes in the input data.

Figure 2.2
Schoolhouse Model Data Flow



RAND *TR378-2.2*

Group-Level Data

This collection of data focuses on higher-level decisions employed in the model. Examples of analyses informed by these data include the following:

- How does the training group treat the absence of noncritical training items? Does the training group delay training, record training deficiencies, or simply work around the missing items?
- What happens when critical training devices are unavailable? Is a training deficiency assigned or does the class wait until an item is available?
- What is the impact of changing the length of the duty day?
- What is the impact of holding classes on Saturdays, Sundays, or both?
- What is the impact of changes to the holiday schedule?

Facility Data

Facilities include classrooms, laboratories, training grounds, ranges, gyms, pools, simulation facilities, dormitories, dining halls, and other location-based resources. Facilities are managed by either a squadron or a group. Squadron-managed facilities are shared among courses within the same squadron but are not available to other squadrons. Group-managed facilities are shared among all squadrons and courses within the group.

Analysis questions involving changes to facility data might include the following:

- What is the impact on training production of changes in the number, availability, or sharing of facilities?
- What is the utilization of current facilities and how does that support future construction requirements?
- As training production is increased, what facilities, if any, create a bottleneck?

Instructor Data

Instructors include all personnel required to provide course training. These include lecturers and individuals needed to safeguard training activities, such as lifeguards, safety personnel, and range supervisors. Instructor data include the number of instructors and their status: civilian or military by grade. Although, the Extend model does not distinguish between military and civilian instructors, nor does it distinguish grade, the data are available to postprocessors for cost computation.

The data input portion of the model also differentiates between certified, basic instructor course (BIC) attendees, and new instructors. However, this differentiation of instructors is not used in the Extend simulation at this time.

Analysis questions involving instructors might include the following:

- What is the impact on instructors when an increase or decrease in the number of additional duties is proposed?
- What is the impact of more or fewer instructors?
- What is the classroom utilization of instructors?

Squadron-Level Data

The squadron-level data include the courses administered by the squadron, the distribution of instructors by course, attrition rates, class start dates, and class sizes. Courses use unique names that must match exactly the corresponding names on the facility sheet, the plans of instruction sheet, and the training devices sheet. The schedule information on the latter half of the squadron sheet is designed to look like program control data (PCD) output from AETC's Training Planning System (TPS).

Analysis decisions at the squadron level might include the following:

- What is the impact of increased or decreased washouts?
- What is the impact of adding classes to the schedule?
- What is the production effect and resource impact of increased class sizes?

Course-Level Data: Plan of Instruction

The POI is the syllabus for each course. The corresponding form in the schoolhouse model defines the number of instructors, the length of instruction, the number and type of training device, and the required facilities. It also includes unique course data, such as overtime waivers and the holding of special training devices over nontraining periods. For example, a course might require overhauling an engine over multiple days and the engine cannot be released to other units until the end of the training sequence. The POI form also adds washback rates at likely points in the POI. The POI form combines information from AETC forms 133, 449, and 896. It most closely resembles AETC form 896 with additional information (washback rates, washback points, nonstandard duty days, and facility requirements). Finally, no AETC form clearly defines facility usage. The POI form makes general reference to facility requirements but does not define the number or specific type in all cases. The POI form adds data columns to standardize the required facility information.

Analysis questions involving the plan of instruction might include the following:

- How do changes in the syllabus affect resource utilization?
- What is the effect of duty-day waivers on course length?
- What is the impact on production of changes in the washback rate or the points of washback?

Course-Level Data: Training Devices

Training devices include all consumable and nonconsumable materials needed for instruction. The training device sheet is a copy of AETC form 120. We appended five additional columns of data for reliability, maintainability, and reparability data.

Analysis choices using the training device data might include the following:

- What are the critical devices needed to teach a course?
- Under what circumstances do training device shortages occur?
- By how much will training deficiencies increase given a reduction in critical training devices or a lack of funding for certain devices?

These six data categories are organized together in the Excel front end. In the next chapter, we walk through the steps necessary to set up and use the front end.

CHAPTER THREE

Front-End User Interface

The front-end user interface of the schoolhouse model utilizes Excel, a commercial spreadsheet program.[1] In this chapter, we discuss in detail the functioning of the front end. The 37th Training Group at Lackland Air Force Base will be used as the exemplar schoolhouse. Because of changes in schoolhouse organization and course POIs for several of the specialties over time, we do not propose that these data be used for analysis. Rather, their use is purely for demonstration purposes. Our discussion begins with the mechanics of starting up the user interface for the first time and proceeds by data type through each of the Excel worksheets.

Initial Start-Up

The first step in using the schoolhouse model is to set up the Extend simulation, Excel front end, and the requisite data files correctly on a computer.[2] The Extend simulation requires a variety of input data files and produces output files that must be stored in specific directory locations. The Excel front end uses Visual Basic routines that also expect to find data files in certain directories. Thus, the first step in implementing the schoolhouse model is to create the proper directory structure and place the appropriate files in the right place.

Locating the Files

The schoolhouse model has three required directories (or locations) for files. Seven required files must be placed correctly into these directories. They are listed in Table 3.1 along with their purposes and where they need to be located. Note that three of the files explicitly carry the current version number of the model. The working directory also carries this designation.

The "Schoolhouse Model setup library v.1.8.7.xls" must be placed in the root directory of the hard drive (e.g., C:\). Two header files, "schoolhouse declare.h" and "schoolhouse declare except Controller.h," are placed in the Extend extensions directory. Additionally,

1 Excel is a part of the Microsoft Office package but can also be purchased separately.

2 This user manual presupposes that the user is operating in a Microsoft Windows® environment. The software should run on a Macintosh® with minor changes in the directory structure, but the authors did not test this capability.

Table 3.1
File Locations for the Schoolhouse Model

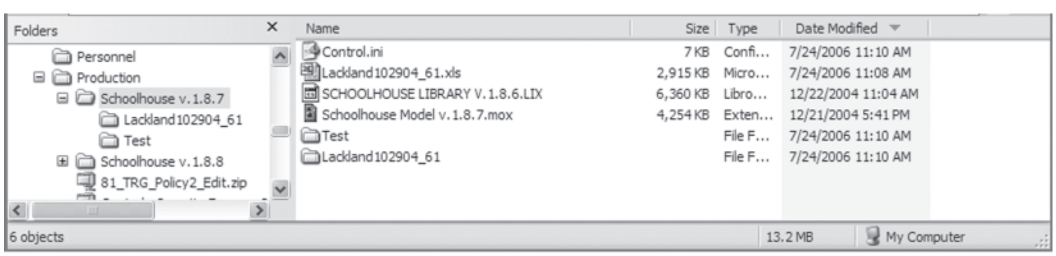
File	Purpose	File Location Directory
Schoolhouse Model setup library v.1.8.7.xls	Required for Excel Visual Basic routines	C:\
schoolhouse declare.h	Header file for array sizes	. . . \Extend\Extensions
schoolhouse declare except Controller.h	Header file	. . . \Extend\Extensions
SchoolTiming.dll	Library file	. . . \Extend\Extensions
<data input Excel file>.xls [can use any name]	Holds all input data	. . . \Production\Schoolhouse v.1.8.7
Schoolhouse library v.1.8.7.lix	Simulation library	. . . \Production\Schoolhouse v.1.8.7
Schoolhouse library v.1.8.7.mox	Simulation code	. . . \Production\Schoolhouse v.1.8.7

NOTE: Ellipses (. . .) represent the path where the directory is found. In most cases, Extend installs itself at the root, so the path for the header files would be C:\Extend\Extensions. The player-only version will install itself in C:\Extend6Player by default.

one dynamic link library, "SchoolTiming.dll," is placed in the Extend extensions directory. Extend creates the extensions directory when it is installed. Both the commercial version and the free run-time module create the same set of required directories.

The last three files referenced in Table 3.1 require a unique directory configuration. The directory they reside in contains the version number of the model within the directory name. The various versions of the model all reside in separate version-number directories under the directory labeled "Production." Scenarios, cases, or different runs of the model will all appear in separate directories below the version-number directory. Figure 3.1 shows an example of this file structure with two different versions of the model (versions 1.8.7 and 1.8.8). Within the version 1.8.7 directory, we have two cases (or scenarios) titled "Test" and "Lackland102904_61." The model places a copy of all data required to run or rerun a case in that case's directory.

Figure 3.1
Master Directory Listing



For example, the "Test" directory has all the text files generated by the Excel front end as well as an additional copy of the Excel input data file. Additionally, there is a "Control.ini" file in the "Test" directory[3] that contains all the information necessary to point the Extend program to the correct directory. To rerun a case, the user must copy the "Control.ini" file from the test directory to the correct version-number directory, which is directly above in the directory structure.[4]

Opening the Excel Front End

The first step in using the model is to open Excel and enable options that will permit running of the Visual Basic macros.[5] The user must first enable the proper security level within Excel. With Excel started, the user selects Tools | Macros | Security. The "Security" dialog box should appear as in Figure 3.2. The user should select "Medium," click "OK," and then quit Excel. This step needs to be performed only once unless the user decides to change the security level at some other time. When that happens, this step needs to be repeated.

With the security level set, an Excel sample file can now be opened. Rather than starting from scratch, we recommend using the sample Excel input file supplied with the model. Go to the model version directory and select the sample Excel input file. Upon starting the Excel file, the user will get an Excel message box as depicted in Figure 3.3. The user should select "Enable Macros."[6]

The introductory screen of the Excel front end titled "Air Force AETC Schoolhouse Model" will appear, as seen in Figure 3.4. The screen contains a number of data items that ensure proper installation of the database, control of the simulation, and access to the data sheets. In order for the front end to function correctly, the correct directory-path names must be entered into the boxes in rows 9 and 13. Row 9 must contain the full path name to the production directory. Row 13 must contain the directory name for the desired model version number. If row 13 does not have a model version number (as in Figure 3.4) but is blank, click on the drop-down list, select a model version, save the file, and then reopen the model. The drop-down list may appear to be blank, but scrolling to the top of the window will reveal a selection of model versions.

---

3 The "Control.ini" file in the above scenario is also copied to the version directory, as seen in Figure 3.1.

4 The MS-DOS® commands would read as follows:

cd [. . .]\production\schoolhouse v.1.8.7 (where [. . .] represents the rest of the directory structure)

copy \Test\Control.ini *.*

Alternatively, the user can highlight "Control.ini" in the "Test" directory, press Ctrl+C (copy), highlight a spot in the version directory, and press Ctrl+V (paste).

"Test" is assumed to be the subdirectory under "Schoolhouse v.1.8.7."

5 We recommend using a sample Excel file already populated with schoolhouse data. The user would then rename the sheets with the required new names.

6 "Lackland102904_61.xls" is the sample Excel input file used in this report.

Figure 3.2
Security Dialog Box

Figure 3.3
Security Warning Message Box

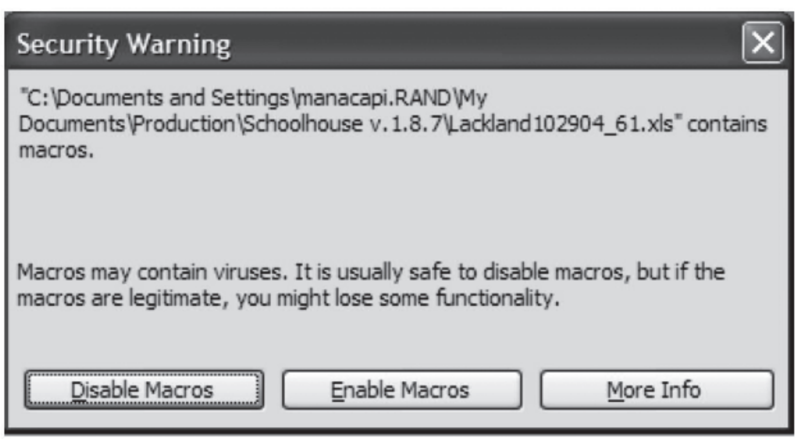


Figure 3.4
Introductory Screen

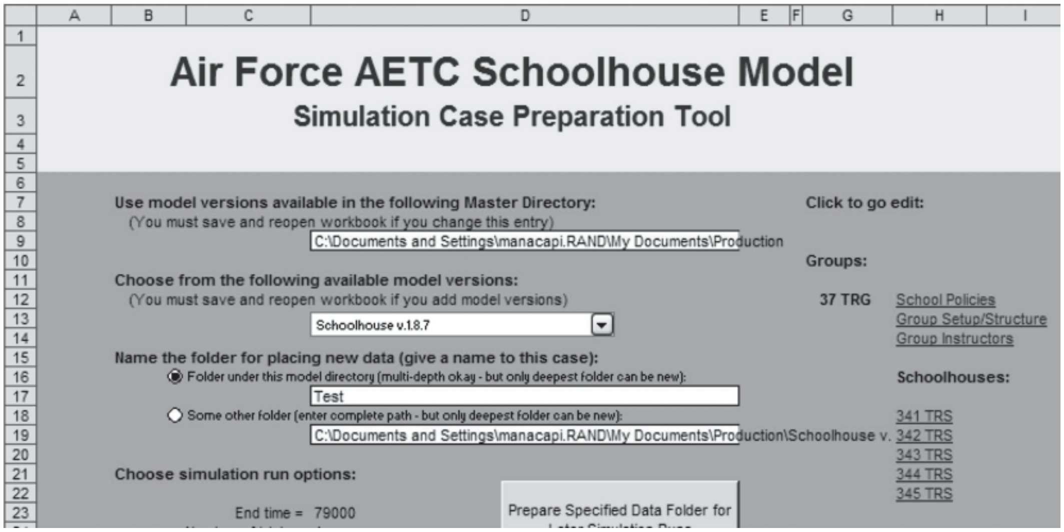
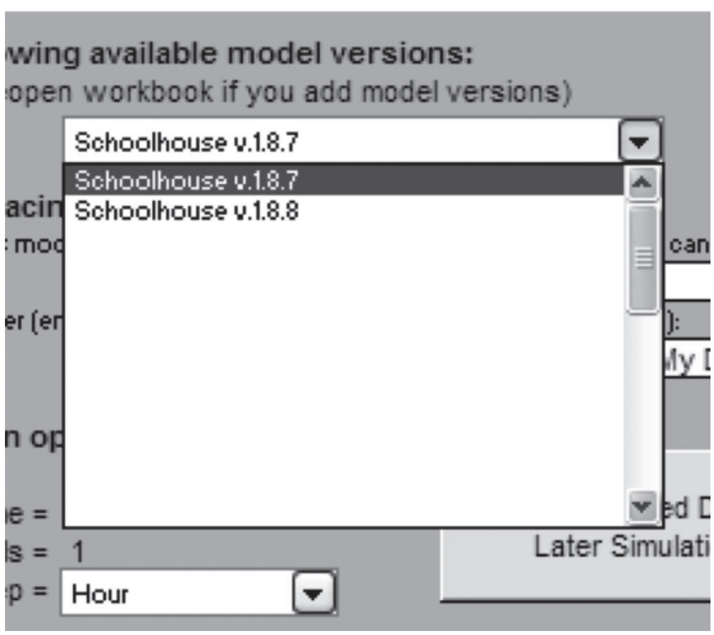


Figure 3.5 depicts an example of what should appear. If the drop-down list is blank even after moving the scroll bar, then the directory structure is not set up as described in Table 3.1. The version names that appear in this drop-down list are taken from the directory names in the "Production" directory.

Figure 3.5
Model Version Selection Drop-Down List

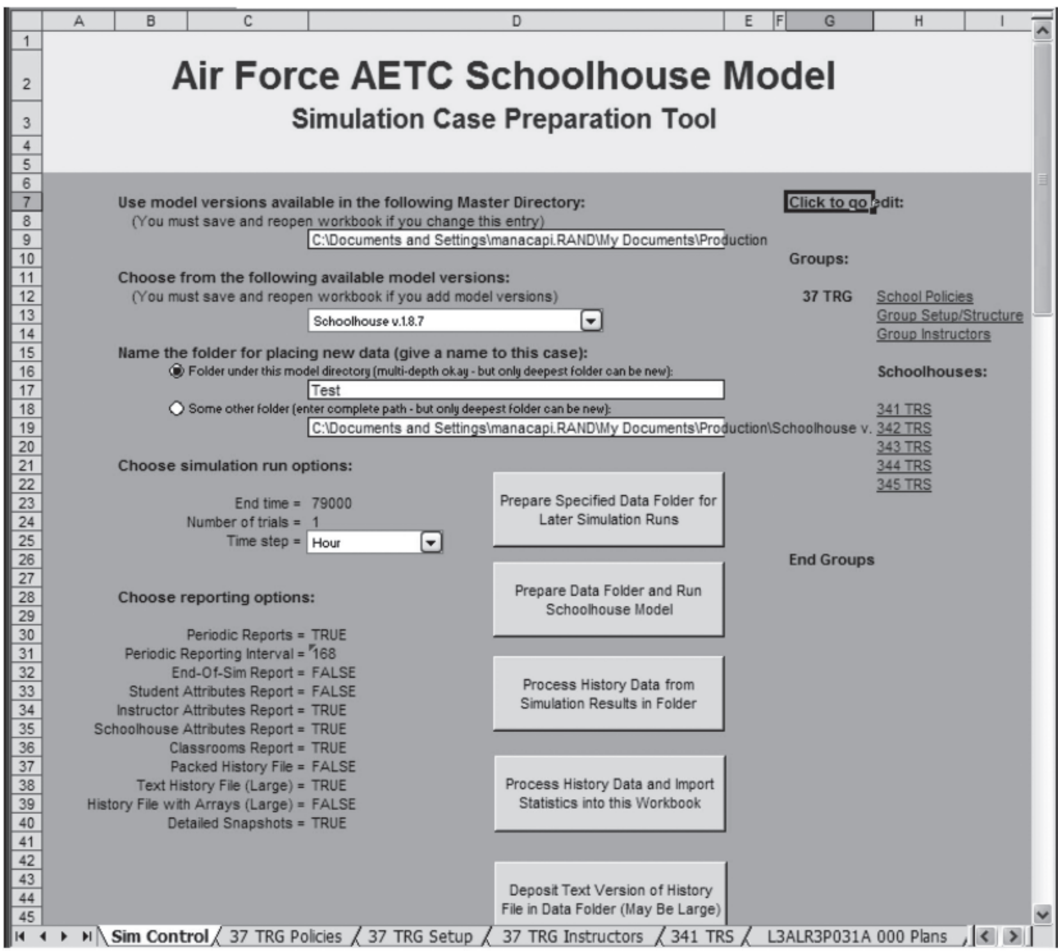


After selecting the model version, save the workbook, close Excel, and reopen the saved workbook to enable these changes. After reopening the Excel sample input file, check that the path and version names are correct. When they are, the front end is correctly configured.

The Introductory Screen

The full introductory screen for the front end is shown in Figure 3.6. Certain text contains hyperlinks. A hyperlink takes the user to a related page. Excel indicates hyperlinks by changing the text color and by underlining the text. In column H in Figure 3.6, there are a

Figure 3.6
Simulation-Control Sheet



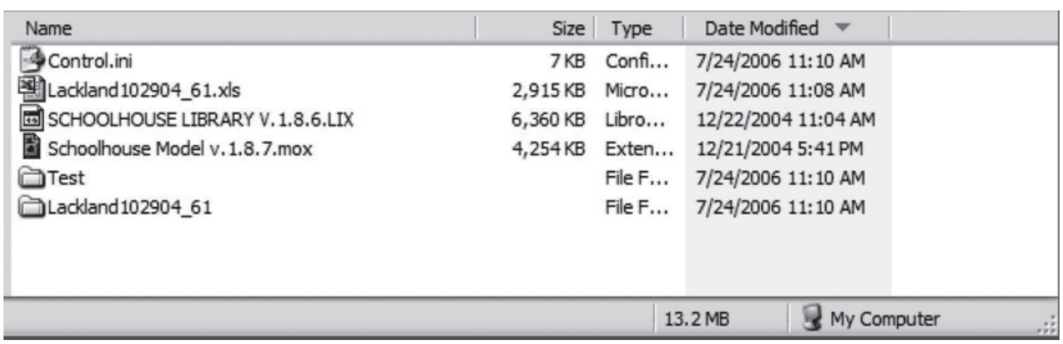
number of useful hyperlinks. As an alternative to using the hyperlink, the user can select the appropriate tab at the bottom of the screen.[7] For example, selecting the "School Policies" hyperlink is the same as selecting the "37 TRG Policies" tab.

Figure 3.6 is referred to as the front page, the introductory screen, or the simulation-control sheet and is the starting point for any changes to the model or scenario. On most of the other sheets of the Excel model, the user will find a button labeled "Return to Sim Control." Clicking this button will return the user to this sheet. We now turn to a description of most of the features and options on this sheet.[8]

Naming the Folder for Placing Data

As discussed previously, setting up the directory structure properly is essential for using the model. All scenario directories reside under the model version directory. For example, in Figure 3.7, the window shows the contents of the "Schoolhouse v.1.8.7" directory.[9] In its directory are two additional directories labeled "Lackland102904_61" and "Test." These two directories contain two separate scenarios. Each directory contains all the input and output files associated with the particular case and run. An unlimited number of scenarios can be placed in any version directory. When the Excel front end creates the input files for an Extend simulation run, it places all the text files in the scenario directory, as well as a copy of the Excel input file and a "Control.ini" file that can be used to rerun that specific scenario.

Figure 3.7
Master Directory Listing



---

7 Note that the sheet names or tab titles must be exact: They must exactly match the corresponding group, squadron, or course name.

8 Not all of the features suggested on the "Sim Control" sheet are implemented. That is also the case with a number of other Excel sheets. These features are included for future implementation.

9 Version 1.8.8 is the latest version of the model as of October 2005.

Save Folder for Current Run

Each run in the model is saved in its own directory within the schoolhouse version directory. Row 17 of the simulation-control page contains a box (Figure 3.8) where the user names the directory for the particular run being built. The user must also select the "Folder under this model directory" radio button. Alternatively, the user can specify another location by filling in the box in row 19 and selecting the "Some other folder" radio button on the simulation-control sheet. In the first case, only the name of the scenario directory is required in the box in line 17. In the second case, the entire directory path to the new scenario directory must be listed in the box on row 19.[10]

Simulation Run Options

The box in cell D23 (see Figure 3.9) specifies the length of each trial of the simulation. The number ("End time") is usually defined in hours. It is possible to use another unit of measurement. To do so, select the new measurement from the drop-down list in row 25.

Cell D24 defines the number of runs or trials. To create multiple replications of the simulation change cell D24 to the desired number of trials. The model will use a different starting random-number seed for each subsequent run or trial.

Reporting Options

None of the reporting options in Figure 3.10 have been implemented. They are potential future implementations to improve the output. Currently, the model produces a large "Text History File" (row 38) for postprocessing and debugging.

Figure 3.8
Labeling the Scenario Directory (Simulation-Control Sheet)

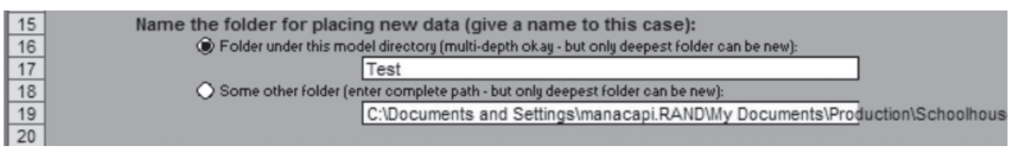
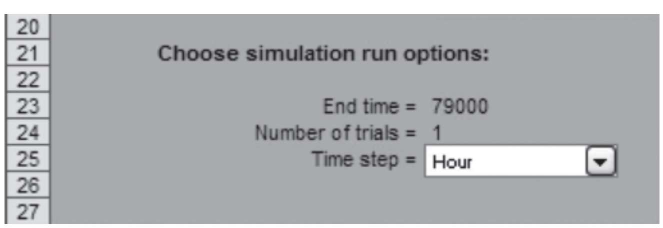


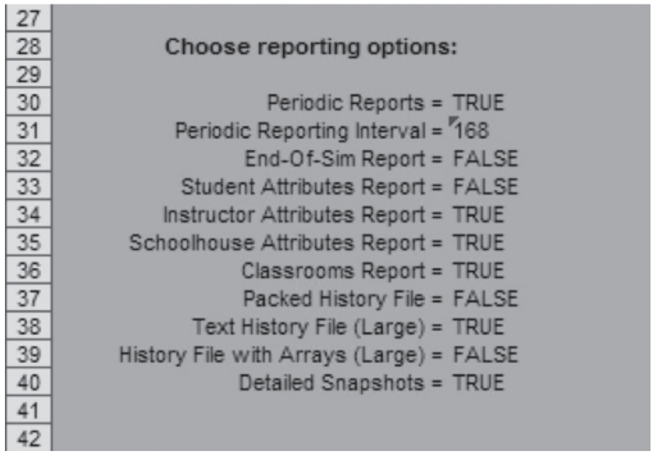
Figure 3.9
Setting Up the Run Time, Replications, and Time Unit (Simulation-Control Sheet)



---

10 In either case, the user does not need to create the directory beforehand; Excel will create the directory upon saving the model data.

Figure 3.10
Reporting Options (Simulation-Control Sheet)



Structure of the Schoolhouse (Right Side of Screen)

The model data are arranged using the Air Force's hierarchical organizational structure: training groups (TRGs) and training squadrons (TRSs). Each TRS represents a schoolhouse. The model is designed to handle one training group. The structure of the group must look like the example in Figure 3.11. The group requires a group name (such as 37 TRG). Next to the group name are links to group-level data items: "School Policies," "Group Setup/Structure," and "Group Instructors." The links direct the user to three corresponding tabs with the following names (where xx represents the group number): "xx TRG Policies," "xx TRG Setup," and "xx TRG Instructors." Under the hyperlinks to the group pages is the keyword "Schoolhouses" followed by links to the individual squadron sheets. Names used to indicate a squadron must match precisely those on the tabs. Finally, the keyword "End Groups" indicates the end of the section.

Buttons

The buttons on the lower half of the main screen (see Figure 3.12) activate macros to manipulate Excel data or output data. Currently, only the top button is implemented. Clicking the "Prepare Specified Data Folder for Later Simulation Runs" button will create a directory in the location specified previously (see Figure 3.8) and build the text input files that Extend requires.

This completes the description of the simulation-control sheet. We now turn to the additional worksheets that define this training scenario.

Figure 3.11
Setting Up the Structure
(Simulation-Control Sheet)

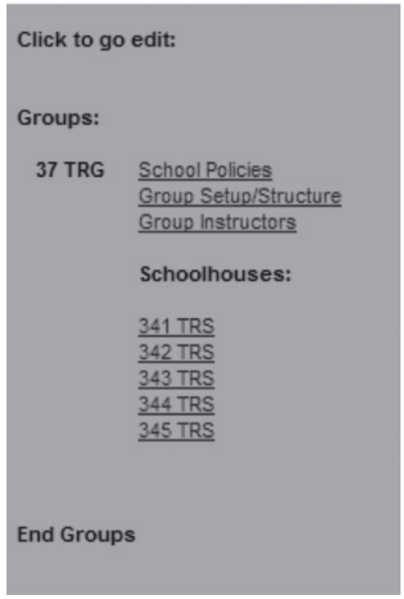
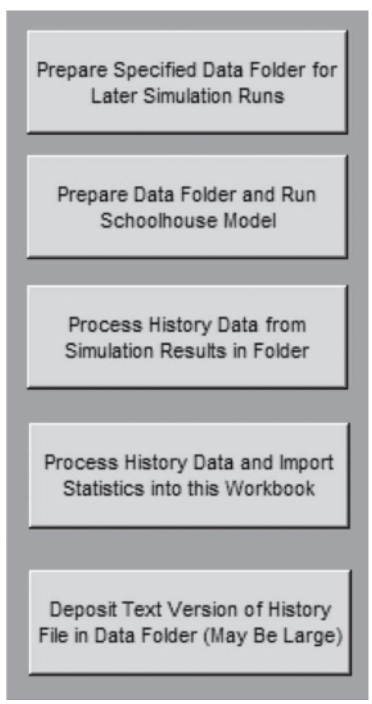


Figure 3.12
Using the Buttons
(Simulation-Control Sheet)



Overview of the Tabs

As previously noted, the tabs follow a strict naming convention (see Figures 3.13 and 3.14). In addition to the simulation-control sheet, each group listed on the front page has three sheets (for policies, setup, and instructors) and each schoolhouse (squadron) has one sheet with the tab name exactly as it appears on the simulation-control sheet. Every course (e.g., L3ALR3P031A 000) will have two sheets (for plans and devices) with the course name exactly as it appears in the schoolhouse sheets and preceding these keywords.

TRG Policy Setup Sheet

The group policy sheet controls a number of key model options. A sample sheet is shown in Figure 3.15. The choices made on the group policy sheet affect all squadrons within that group. The sheet is divided into five main blocks, each of which contains decisions, policy choices, and data.

Resources and Training Devices Block

Training devices are an important resource for successful training. In the model, the user assigns all devices to one of four categories, labeled 0, 1, 2, and 3 (see the section TRS Course Resource Sheet, later in this chapter). Training devices assigned to category 0 are ignored. Training devices assigned to category 1 are not modeled but can be tracked by postprocessing the history file. Training devices assigned to category 2 are modeled and considered important but, if not available, the instruction can still be delivered. Training devices assigned to category 3 are critical and if not available will result in either the training being delayed or a training deficiency assigned to members of the class.

Figure 3.13
Leftmost Tabs of the Excel Case File

Figure 3.14
Rightmost Tabs of the Excel Case File

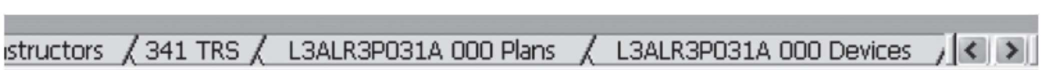
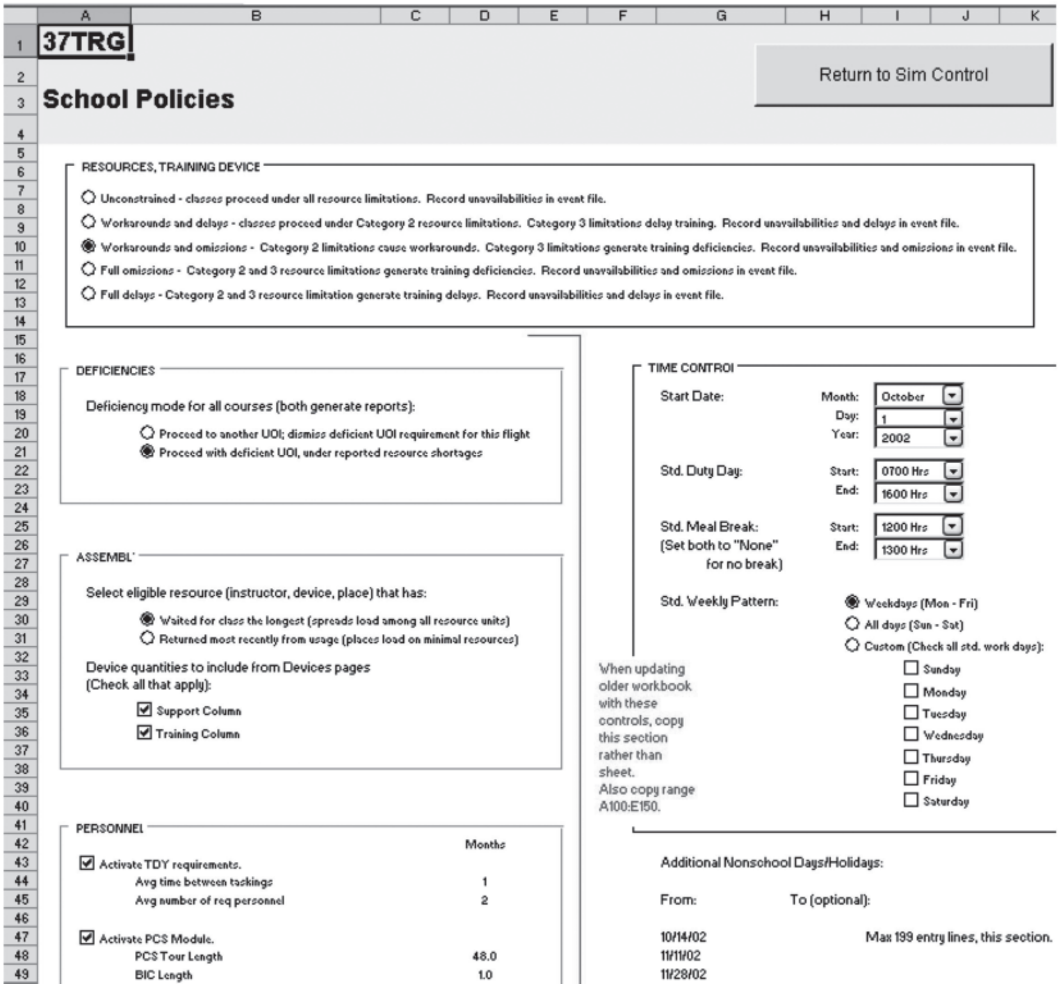


Figure 3.15
Policy Setup Sheet

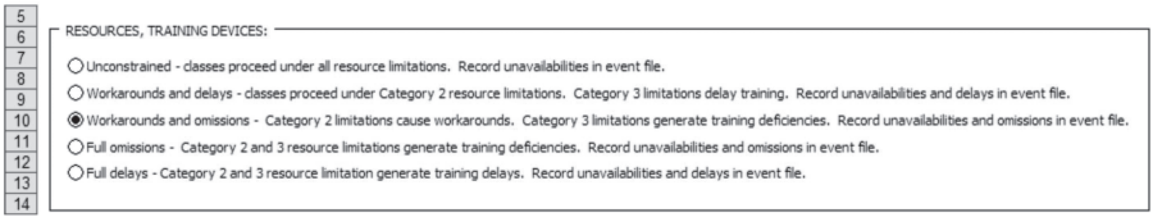


The resources and training devices block (see Figure 3.16) allows the user to select from five different options that will determine how limited resources in various categories affect the model outcomes. Resources affected by this choice include facilities, instructors, and training devices. The first option, "Unconstrained," essentially ignores any resource shortages. This option should provide the required number of trainees in the least time possible.

The second option, "Workarounds and delays," ignores category 2 shortages, but category 3 shortages cause a delay until the resources become available. This option ignores any interventions that could make up for or reduce the delay caused by a shortage.

The third option, "Workarounds and omissions," is the standard choice. Category 2 shortages are worked around and category 3 shortages generate training deficiencies for the flight.

Figure 3.16
Schoolhouse Model Options (Policy Setup Sheet)



The fourth option, “Full omissions,” generates training deficiencies for any category 2 or category 3 shortages.[11] This option is a more accurate measure of the impact of insufficient funding for training devices. The impact that it imposes on the training product, the trainee, is more representative of what actually happens when a required training device is not available when it is needed.

The final option, “Full delays,” measures the impact of delaying training until all required devices become available.

Deficiencies Block

The deficiencies block (see Figure 3.17) identifies the course of action when deficiencies are created. Two options are currently available. The first is to skip a unit of instruction (UOI) because of a missing training device and proceed with the next UOI. The second option will cause a training deficiency to be noted and the training to be continued with the current UOI.

For example, suppose a category 3 training device is unavailable for the second UOI during a run with the workarounds and omissions option enabled. The first deficiencies option causes the model to skip to the third UOI. The second continues with the second UOI. In both cases, a deficiency is recorded for each flight.

Assembly Block

The assembly block (see Figure 3.18) has two groups of options. The first group determines how resources are selected. It contains two choices. The first provides needed resources by choosing the resource that is least utilized. The second provides resources based on those most recently used. The top selection performs like a last-in-first-out queue and spreads the utilization among all instructors, training devices, and facilities. The other selection is essentially a first-in-first-out queue that maximizes the utilization of the fewest resources. This latter option is useful in that it gives some immediate insight into the fewest instructors, training devices, or facilities required before generating deficiencies or delays.

---

[11] A training deficiency is recorded on an individual’s military training record, noting that an individual missed some instruction normally required for graduation. The training item will be taught at the individual’s first assignment.

Figure 3.17
Deficiency Options (Policy Setup Sheet)

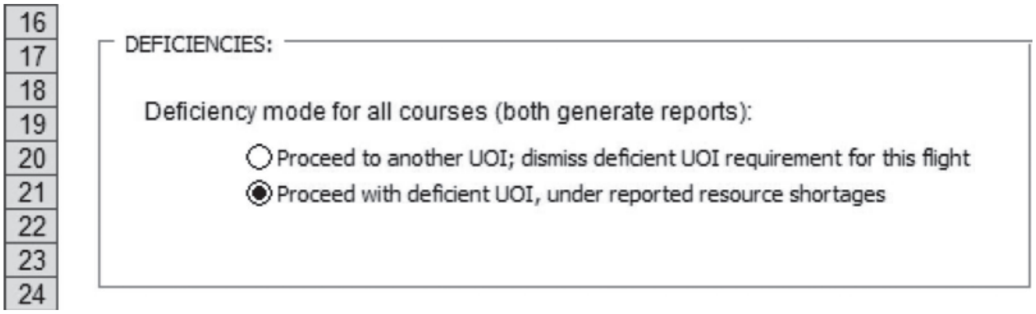
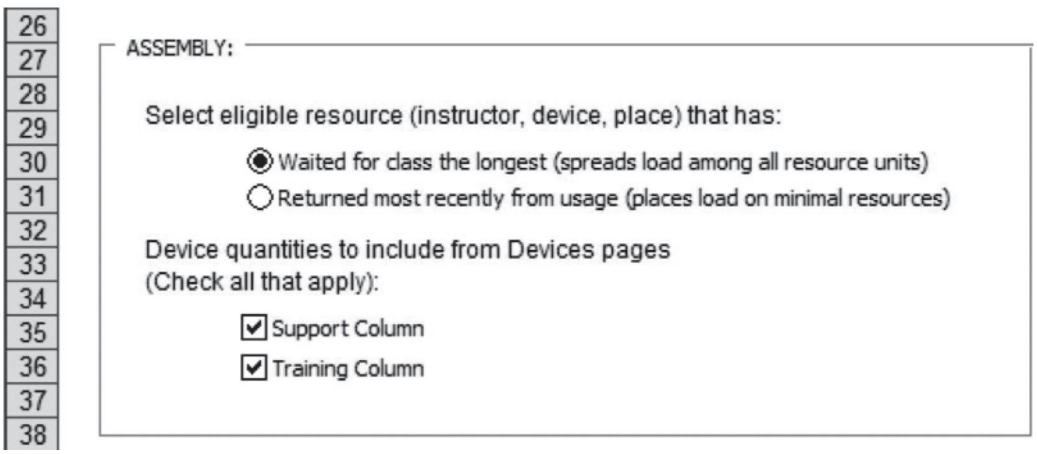


Figure 3.18
Resource Options (Policy Setup Sheet)

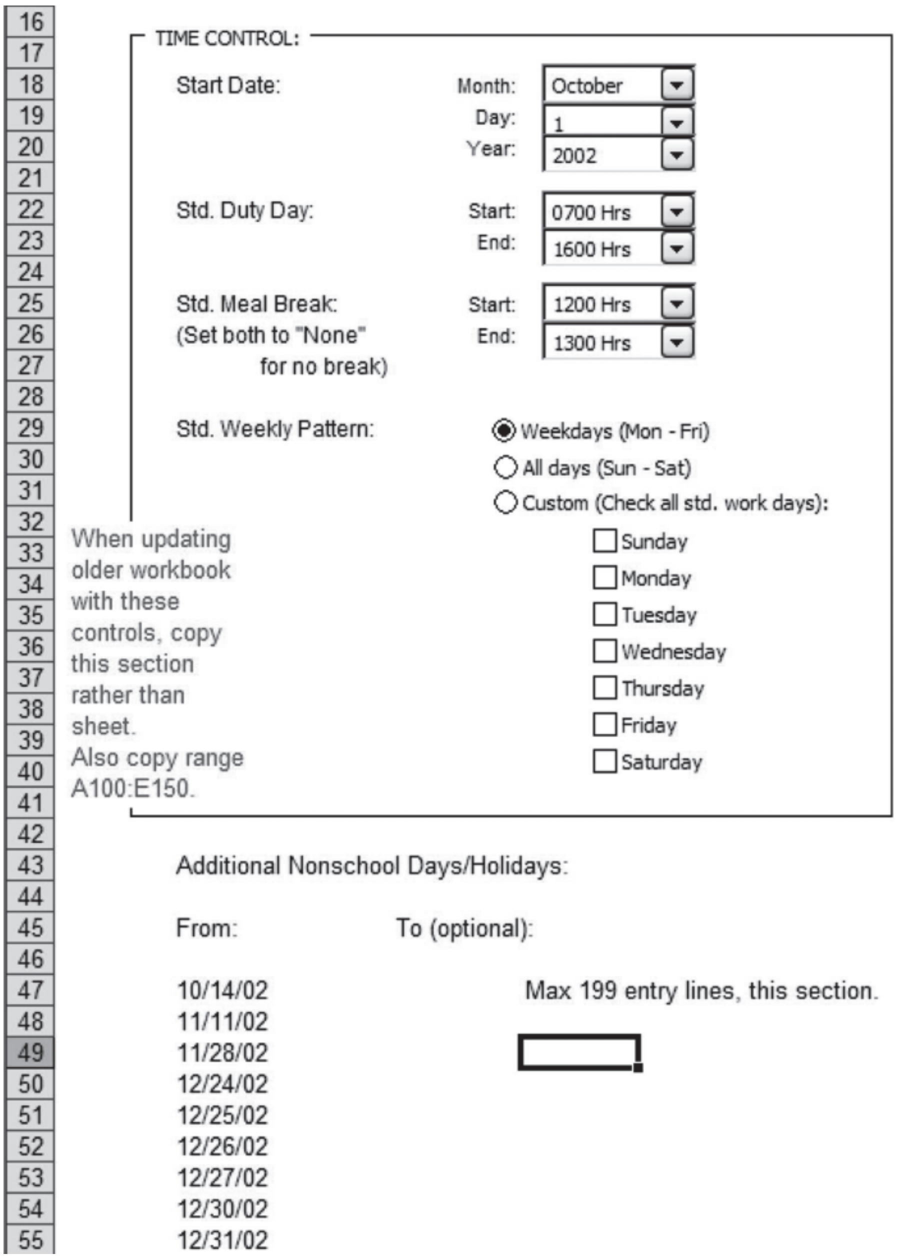


The second group in the assembly block determines how the total number of available devices is calculated. Typically, the training column represents the devices that are available for training and the support column represents the additional average number of devices in maintenance. In most cases, only the training column will be checked. This means that only the average number of available training devices for instruction will be counted in the overall total. The typical choice is whether to also include the number of training devices listed in the support column. A more detailed discussion follows later in this chapter.

Time-Control Block

The time-control block contains five major pieces of information (see Figure 3.19). The first section of the block determines the starting date of the model. The drop-down lists are self-explanatory. Class schedules with start dates prior to the start date of the model will generate a warning message and will need to be corrected. The second section defines the duty day in terms of hours of class instruction, not counting meal breaks. The model will fill in the hours of instruction before breaking for the day.[12] The third set of boxes defines a meal break during which no instruction occurs. The fourth part of this block defines the standard workweek.

Figure 3.19
Work Shift and Holiday Input (Policy Setup Sheet)



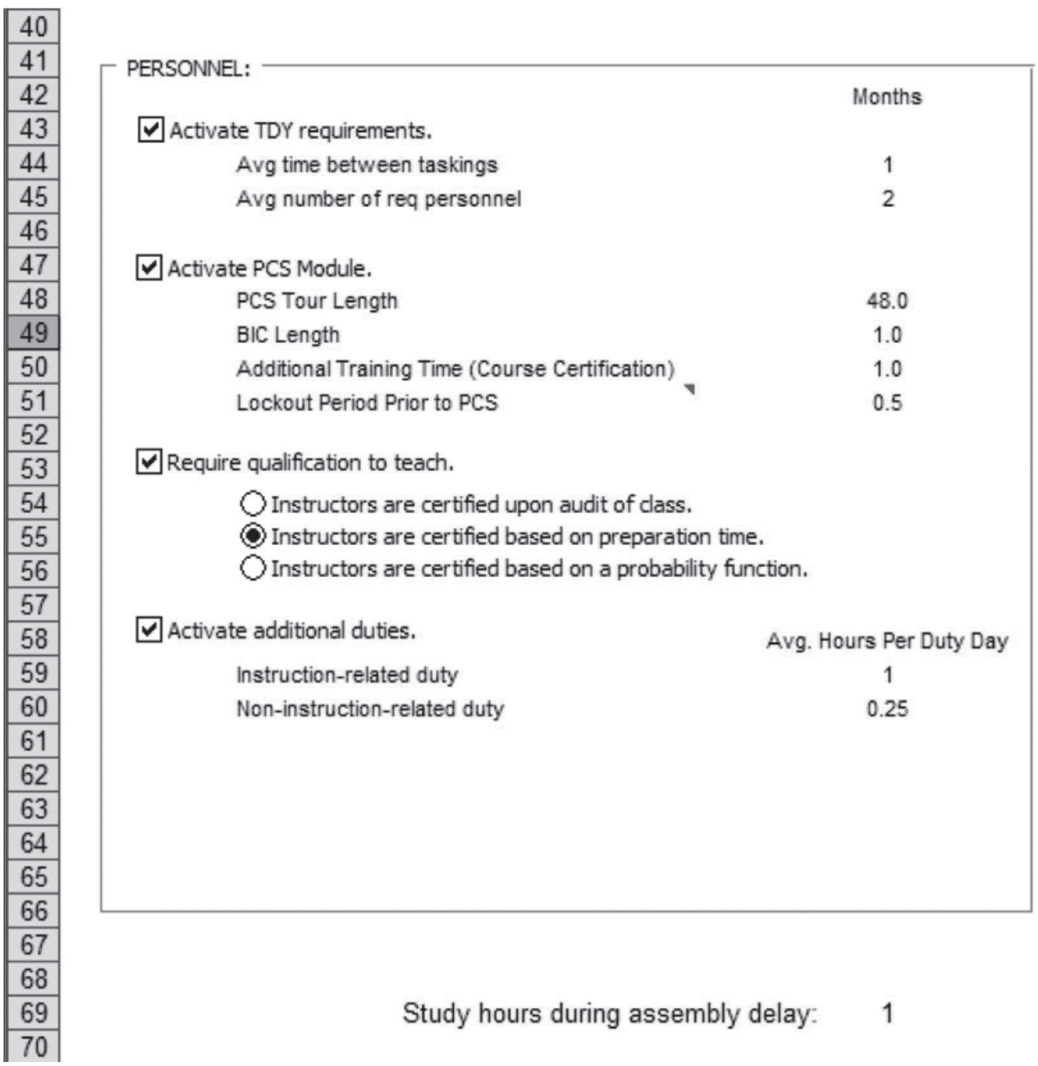
[12] It is possible to extend beyond the duty day; that information is input directly into the POI sheet as described in the section on schedule data, later in this chapter.

Finally, the last part of this block defines holidays and nontraining days. The model will not perform instruction on the days listed in this section of the block. This section is used to define government holidays.

Personnel Block

The personnel block is the final input section on this sheet (see Figure 3.20). None of the features in the personnel block are currently implemented in the model. They are provided as a placeholder for future implementation plans that will examine resource options for instructors and staff in more detail.

Figure 3.20
Miscellaneous Personnel Options (Policy Setup Sheet)



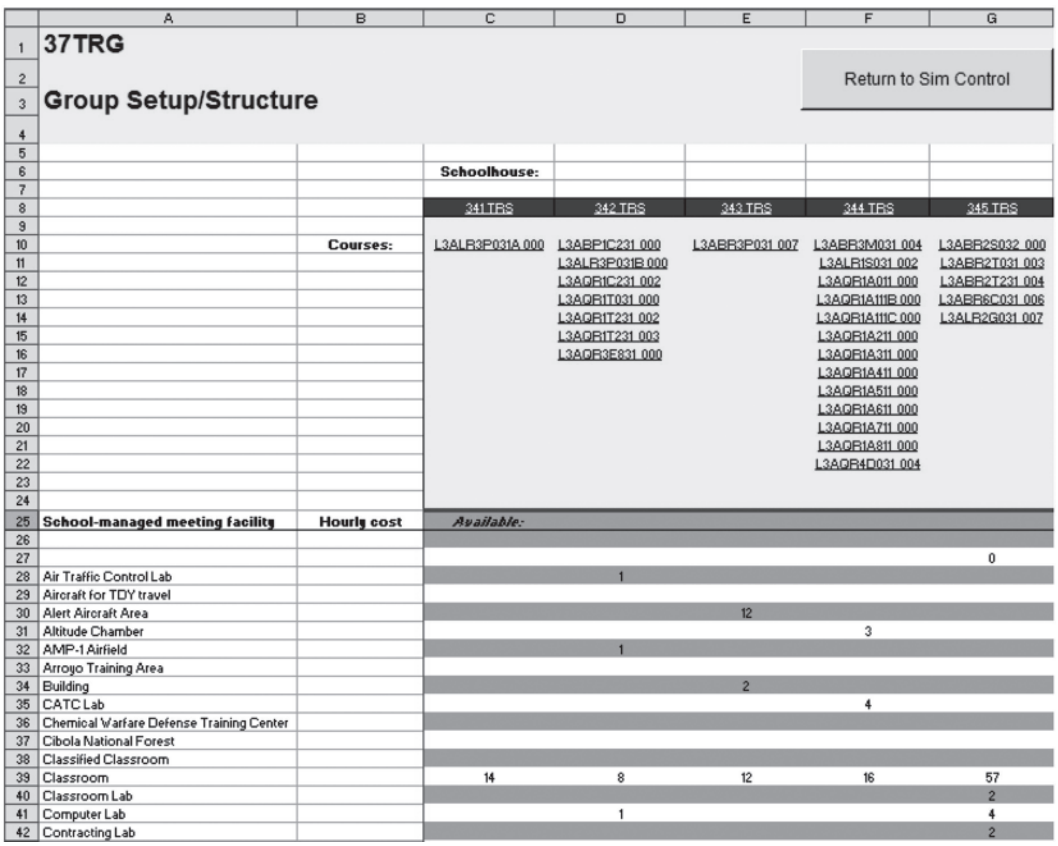
TRG Setup/Structure Sheet

The second tab for the training group is the TRG setup/structure sheet (Figure 3.21) that defines the organizational structure of the training squadrons and their associated classes, the classroom types, the number of each type of classroom, and the owner of the classroom resource. In this example, there are five training squadrons, numbered 341, 342, 343, 344, and 345. Each of these squadrons conducts one or more courses and is assigned multiple facilities.

Courses

Squadron names are listed in row 8 of the TRG setup/structure sheet, as shown in Figure 3.21. Courses provided by each squadron are listed below the squadron name (see Figure 3.22). For example, course L3ALR3P031A 000[13] is provided by the 341st TRS, and the 342nd TRS is

Figure 3.21
TRG Setup/Structure Sheet



	A	B	C	D	E	F	G
1	37TRG						
2						Return to Sim Control	
3	Group Setup/Structure						
4							
5							
6			Schoolhouse:				
7							
8			341 TRS	342 TRS	343 TRS	344 TRS	345 TRS
9							
10		Courses:	L3ALR3P031A 000	L3ABP1C231 000	L3ABR3P031 007	L3ABR3M031 004	L3ABR2S032 000
11				L3ALR3P031B 000		L3ALR1S031 002	L3ABR2T031 003
12				L3AQR1C231 002		L3AQR1A011 000	L3ABR2T231 004
13				L3AQR1T031 000		L3AQR1A111B 000	L3ABR6C031 006
14				L3AQR1T231 002		L3AQR1A111C 000	L3ALR2G031 007
15				L3AQR1T231 003		L3AQR1A211 000	
16				L3AQR3E831 000		L3AQR1A311 000	
17						L3AQR1A411 000	
18						L3AQR1A511 000	
19						L3AQR1A611 000	
20						L3AQR1A711 000	
21						L3AQR1A811 000	
22						L3AQR4D031 004	
23							
24							
25	School-managed meeting facility	Hourly cost	*Available:*				
26							
27							0
28	Air Traffic Control Lab			1			
29	Aircraft for TDY travel						
30	Alert Aircraft Area				12		
31	Altitude Chamber					3	
32	AMP-1 Airfield			1			
33	Arroyo Training Area						
34	Building				2		
35	CATC Lab					4	
36	Chemical Warfare Defense Training Center						
37	Cibola National Forest						
38	Classified Classroom						
39	Classroom		14	8	12	16	57
40	Classroom Lab						2
41	Computer Lab			1			4
42	Contracting Lab						2

[13] AETC uses two spaces in the course name separating the course number from the three-digit version number.

Figure 3.22
Course Listing (TRG Setup/Structure Sheet)

	B	C	D
5			
6		**Schoolhouse:**	
7			
8		341 TRS	342 TRS
9			
10	**Courses:**	L3ALR3P031A 000	L3ABP1C231 000
11			L3ALR3P031B 000
12			L3AQR1C231 002
13			L3AQR1T031 000
14			L3AQR1T231 002
15			L3AQR1T231 003
16			L3AQR3E831 000
17			
18			
19			
20			

responsible for seven different courses. The course names are the official AETC designations. These names must match the names on the associated sheets that contain more detailed course-related information.

School-Managed Meeting Facilities

Below the list of squadrons and courses, in the same column as the course listing, is the number of facilities by type operated by the squadron (see Figure 3.23). The facility type is listed on the left. For example, the air traffic control laboratory belongs to the 342nd TRS and there is exactly one of them. The 342nd also controls one AMP-1 airfield, eight classrooms, and one computer lab. The facilities are listed in alphabetical order. Only a portion of them is shown in Figure 3.23. To determine the full list of associated facilities, use the window scroll bar to see the entire list. These facility types are unique in name and are not shareable between squadrons. They are shareable only among courses in the same squadron.

There is also a column for hourly cost. The model does not use the hourly cost data but they are available in the text files for use in postprocessors.

Group-Managed Meeting Facilities

Farther down on the sheet is a list of group-owned facilities (see Figure 3.24). This section is differentiated by the heading "Group-managed meeting facility." For example, there are four base gyms owned by the 37th TRG. These facilities are available to any course in any squadron within the group.

Other Facilities

The last section of the TRG setup/structure sheet is not implemented. This section defines additional facilities, such as dorm rooms and dining halls (see Figure 3.25). Costs for these facilities may be specified as a one-time fixed cost or as incremental costs. The data are available to postprocessors but are not used in the model.

Figure 3.23
Facilities Available (TRG Setup/Structure Sheet)

	A	B	C	D
8			341 TRS	342 TRS
9				
10		**Courses:**	L3ALR3P031A 000	L3ABP1C231 000
11				L3ALR3P031B 000
12				L3AQR1C231 002
13				L3AQR1T031 000
14				L3AQR1T231 002
15				L3AQR1T231 003
16				L3AQR3E831 000
17				
18				
19				
20				
21				
22				
23				
24				
25	**School-managed meeting facility**	**Hourly cost**	***Available:***	
26				
27				
28	Air Traffic Control Lab			1
29	Aircraft for TDY travel			
30	Alert Aircraft Area			
31	Altitude Chamber			
32	AMP-1 Airfield			1
33	Arroyo Training Area			
34	Building			
35	CATC Lab			
36	Chemical Warfare Defense Training Center			
37	Cibola National Forest			
38	Classified Classroom			
39	Classroom		14	8
40	Classroom Lab			

Figure 3.24
Group Facilities Available (TRG Setup/Structure Sheet)

89			
90	**Group-managed meeting facility**	**Hourly cost**	**Available**
91	Base Gym		1
92	Base Swimming Pool		2
93			
94			

Figure 3.25
Other Facility Input (TRG Setup/Structure Sheet)

104				
105	**Other group facility**	**Hourly cost**	**Cost (one-time)**	**Available**
106	Dormitory A			
107	Dormitory B			
108				
109				
110	**Other costs**	**Yearly cost**	**Hourly cost**	**One-time cost**
111				
112				
113	Instructor pay			
114				

TRG Instructor Summary Sheet

The group's instructor summary sheet (see Figure 3.26) is critical to the function of the model. Values are output from this page into the text file used by the Extend model. However, the numbers of instructors shown in the matrix on this sheet are not entered here. Rather, these data are automatically updated from the course entries on the squadron sheet (see Figure 3.30).

Other data shown on this sheet are for a potential capability to more precisely track instructor grade, cost, and preparation. This sheet would account for instructors requiring completion of the BIC and certification to teach a specific course (usually done through auditing an instructor's current course). Additionally, the model does not currently differentiate by grade; this is another feature reserved for future implementation. The instructor cost data, which may be input on this sheet, are output to text files for postprocessing but are not used by the Extend simulation.

TRS Sheet

Each squadron (schoolhouse) has a squadron sheet like the one shown in Figure 3.27. The squadron sheet contains four broad areas of data: (1) course information, (2) operational information, (3) the total number of instructors, and (4) course schedules (not shown in Figure 3.27).

Course Data

The first major area of data is the initial course data (see Figure 3.28). Each nonblank line represents a new course. For example, the first course, L3ABP1C231 000, is the combat controller apprentice course. It lasts 61 days and has a personnel data system (PDS) code 319. The course number (or name) must be unique and must match exactly the course plans sheet and the course devices sheet. The course title, number of days, PDS code, training manager (TM), and program manager (PM) fields are informational only and are not used by the model. They are output into the text files as needed for postprocessing.

Figure 3.26
TRG Instructor Summary Sheet

37TRG

Instructor Summary

Return to Sim Control

Instructor Costs and Numbers, by Squadron and Grade

	Yearly Cost (per Instr)	Hourly Cost (per Instr)	341 TRS	342 TRS	343 TRS	344 TRS	345 TRS				Total
Course-Certified											
GS11	$67,923	$23.67	11	47	163	23	57				301
GS12	$81,405	$28.36									0
E3	$37,304	$13									0
E4	$43,261	$15									0
E5	$52,695	$18.36									0
E6	$62,174	$21.66									0
E7	$71,334	$24.86									0
Total			11	47	163	23	57	0	0	0	301
BIC Completed / Not Certified											
GS11	$67,923	$23.67									0
GS12	$81,405	$28.36									0
E3	$37,304	$13.00									0
E4	$43,261	$15.07									0
E5	$52,695	$18.36									0
E6	$62,174	$21.66									0
E7	$71,334	$24.86									0
Total			0	0	0	0	0	0	0	0	0
No BIC or Certification											
GS11	$67,923	$23.67									0
GS12	$81,405	$28.36									0
E3	$37,304	$13.00									0
E4	$43,261	$15.07									0
E5	$52,695	$18.36									0
E6	$62,174	$21.66									0
E7	$71,334	$24.86									0
Total			0	0	0	0	0	0	0	0	0
Total	**(Weighted)**	**(Weighted)**									
GS11	$67,923	$23.67	11	47	163	23	57	0	0	0	301
GS12	$81,405	$28.36	0	0	0	0	0	0	0	0	0
E3	$37,304	$13.00	0	0	0	0	0	0	0	0	0
E4	$43,261	$15.07	0	0	0	0	0	0	0	0	0
E5	$52,695	$18.36	0	0	0	0	0	0	0	0	0
E6	$62,174	$21.66	0	0	0	0	0	0	0	0	0
E7	$71,334	$24.86	0	0	0	0	0	0	0	0	0
Total			11	47	163	23	57	0	0	0	301

Sim Control / 37 TRG Policies / 37 TRG Setup / **37 TRG Instructors** / 341 TRS

Figure 3.27
Squadron Sheet

342 TRS

Return to Sim Control

Schoolhouse Summary

Course					
Course Number	Course Title	# Days	PDS	TM	PM
L3ABP1C231 000	COMBAT CONTROL APPRENTICE	61	319		
L3ALR3P031B 000	COMBAT ARMS APPRENTICE COURSE	45	5SI		
L3AQR1C231 002	Combat Control Orientation Course	10	OCR		
L3AQR1T031 000	Survival Evasion Resistance Escape Prelimi	9	LBU		
L3AQR1T231 002	Pararescue Indoctrination Course	49	PJI		
L3AQR1T231 003	Pararescue Preparatory Course	10	P55		
L3AQR3E831 000	EXPLOS ORDNANCE DISP PRELIMINARY	6	2CV		

	Operation Levels											Certified Instructors, by Grade							
Course Number	Groups	Flight Mean Size	Flight Size Std	Min Flight	Max Flight	Total Entries	Rooms Added	Avg. Washback	Program Elimination	Current Elimination	Repeat Interval (hours)	GS11	GS12	E3	E4	E5	E6	E7	Total
L3ABP1C231 000	61	15	12.73	6	24			0.0%		0.0%	8	12							12
L3ALR3P031B 000	194	12	5.66	8	16			0.0%		0.0%	8	13							13
L3AQR1C231 002	54	17.5	17.68	5	30			0.0%		0.0%	8	4							4
L3AQR1T031 000	64	14	14.14	4	24			0.0%		0.0%	8	2							2
L3AQR1T231 002	16	40	42.43	10	70			0.0%		0.0%	8	9							9
L3AQR1T231 003	16	40	42.43	10	70			0.0%		0.0%	8	1							1
L3AQR3E831 000	274	12	11.31	4	20			0.0%		0.0%	8	6							6

Figure 3.28
Squadron Course Data (Squadron Sheet)

342 TRS

Schoolhouse Summary

Course					
Course Number	Course Title	# Days	PDS	TM	PM
L3ABP1C231 000	COMBAT CONTROL APPRENTICE	61	319		
L3ALR3P031B 000	COMBAT ARMS APPRENTICE COURSE	45	5SI		
L3AQR1C231 002	Combat Control Orientation Course	10	OCR		
L3AQR1T031 000	Survival Evasion Resistance Escape Preliminary C	9	LBU		
L3AQR1T231 002	Pararescue Indoctrination Course	49	PJI		
L3AQR1T231 003	Pararescue Preparatory Course	10	P55		
L3AQR3E831 000	EXPLOS ORDNANCE DISP PRELIMINARY	6	2CV		

Operational-Level Data

The second major area of data is the operational data (see Figure 3.29). The first course, L3ABP1C231 000, is 61 days in length with a mean flight size of 15 and a standard deviation of 12.73. The minimum class size is six students and the maximum is 24. Current elimination is the attrition rate and is given here at 0 percent. A new course is started every eight duty-hours, the repeat interval. Currently, the model uses only the maximum flight size and current elimination columns. The other information was used in an earlier version of the model before detailed scheduling was added. The actual flight size and standard deviation for a flight

Figure 3.29
Additional Squadron Course Data (Squadron Sheet)

A	G	H	I	J	K	L	M	N	O	P	Q
342 TRS	Return to Sim Control										
Schoolhou											
Course	**Operation Levels**										
Course Number	Groups	Flight Mean Size	Flight Size Std Dev	Min Flight	Max Flight	Total Entries	Rooms Added	Avg. Washback	Program Elimination	Current Elimination	Repeat Interval (hours)
L3ABP1C231 000	61	15	12.7	6	24			0.0%		0.0%	8
L3ALR3P031B 000	194	12	5.7	8	16			0.0%		0.0%	8
L3AQR1C231 002	54	17.5	17.7	5	30			0.0%		0.0%	8
L3AQR1T031 000	64	14	14.1	4	24			0.0%		0.0%	8
L3AQR1T231 002	16	40	42.4	10	70			0.0%		0.0%	8
L3AQR1T231 003	16	40	42.4	10	70			0.0%		0.0%	8
L3AQR3E831 000	274	12	11.3	4	20			0.0%		0.0%	8

are recorded for each new flight in the scheduling portion of the data (see Figure 3.31). The current elimination column supplies the class attrition rate for all flights taking that course. Washback rates are included in the actual plan of instruction (see Figure 3.32).

Certified Instructors by Grade Data

The third major area of data on the squadron sheet is the number of certified instructors (see Figure 3.30). This table contains the number of instructors for each course. For course L3ABP1C231 000, for example, there are 12 GS-11–level instructors. Currently, the model does not differentiate by grade, although the grade data can be used in the postprocessors to compute costs. The grade structure is designed as a potential future upgrade to the model.

Schedule Data

The last area of data on the squadron sheet lists the course schedules (see Figure 3.31). Course schedules appear much farther down on the sheet. In the example below, the schedule starts on row 55. Course schedules do not have to begin on row 55 of the squadron sheet, however. The model searches for the keyword "course schedules" in column G of the squadron sheet.

Each group of classes for a specific course repeats the information in bold type in rows 57–61 of Figure 3.31. Course-specific information is not in bold type. For example, the keyword "Course" indicates a new group of classes for a particular course, usually a fiscal year's worth, though it is not limited to any time frame. Next to the word "Course" is the specific course number, in this case L3ABP1C231 000, which must match one of the course numbers listed above on the sheet (see Figure 3.28).

Figure 3.30
Squadron Course Instructors (Squadron Sheet)

A	R	S	T	U	V	W	X	Y
342 TRS								
Schoolhou								
Course	**Certified Instructors, by Grade**							
Course Number	GS11	GS12	E3	E4	E5	E6	E7	Total
L3ABP1C231 000	12							12
L3ALR3P031B 000	13							13
L3AQR1C231 002	4							4
L3AQR1T031 000	2							2
L3AQR1T231 002	9							9
L3AQR1T231 003	1							1
L3AQR3E831 000	6							6

This part of the sheet is intended to look exactly like output from AETC's TPS. Consequently, much of the data is not relevant to the model (for example columns O through W) but makes transferring from TPS to the model easier. We added an extra column (column M, "Std Dev" or standard deviation) and changed the title of column L (to "Size/Mean") to accommodate AETC requests. The model does not use all the data from TPS but does use start date, size/mean, and standard deviation. For example, the first class for this course (listed as 1 under "CL/NR," column G) starts on December 2, 2003, and ends on April 26, 2004, with a capacity of 24 trainees. The front end will continue reading through this list of courses and classes until it encounters the 50th blank line. At that point, it is assumed that the schedule data have been exhausted. Although the TPS data contain the number of trainees by TRQI (training resource quantity indicator) code (e.g., AMD0, indicating lateral trainees in Figure 3.31), the simulation does not use this information.

Two or more classes of the same course can start on the same date and the simulation will model the classes as different entities.

TRS Course Plans of Instruction Sheet

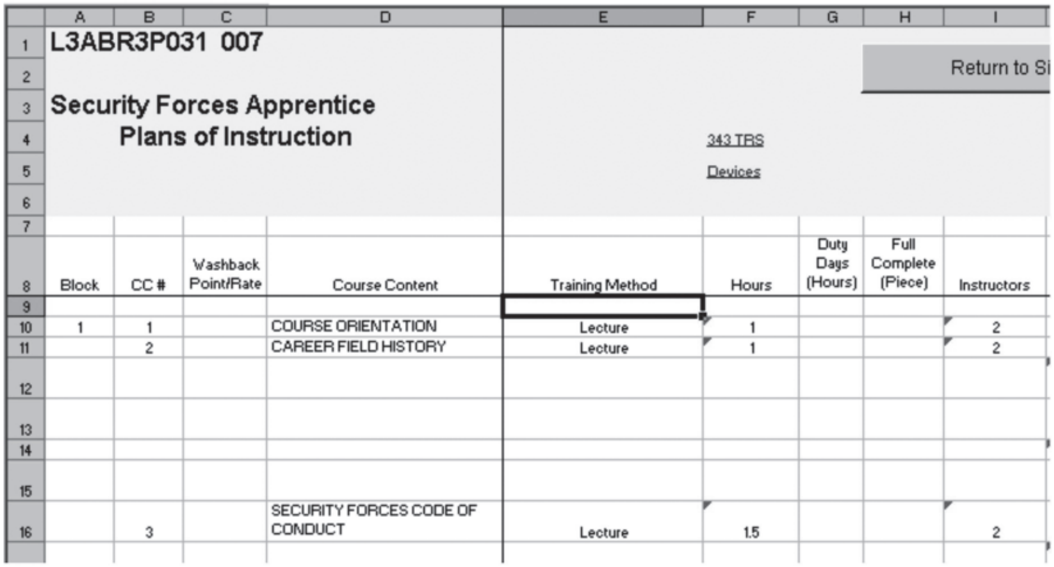
The plans of instruction sheet most closely resembles AETC form 896, but it also contains information from AETC forms 133 and 449 (see Figures 3.32 through 3.35). Additionally, washback rates and facility requirements are included on this sheet. Figure 3.32 shows the top third of the sheet.

Figure 3.31
Course Schedule Data (Squadron Sheet)

	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
54																	
55	**Course Schedules**																
56																	
57	**Course:**	L3ABP1C231 000						**FY:**	4		**Sequence:**		PM		**PM:**		7
58																	
59										**TDY**			**NTDY**			**PP**	
60																	
61	**CL/NR**	**Start Dt**	**Grad Dt**	**TM**	**TLC**	**Size/ Mean**	**Std Dev**	**NR/ GPS**	**Off**	**Amn**	**Civ**	**Off**	**Amn**	**Civ**	**Off**	**Amn**	**Civ**
62	1	12/2/2003	4/26/2004			24	0										
63								AMD0		126							
64	2	1/22/2004	6/7/2004			24	0										
65								AMD0		126							
66	3	3/3/2004	7/16/2004			24	0										
67								AMD0		126							
68	4	5/3/2004	9/16/2004			24	0										
69								AMD0		126							
70	5	7/1/2004	11/17/2004			24	0										
71								AMD0		126							
72	6	8/12/2004	1/6/2005			24	0										
73								AMD0		126							
74	7	9/22/2004	2/16/2005			24	0										
75								AMD0		126							
76																	
77																	
78	**Course:**	L3ABP1C231 000						**FY:**	5		**Sequence:**		PM		**PM:**		7
79																	
80										**TDY**			**NTDY**			**PP**	
81																	
82	**CL/NR**	**Start Dt**	**Grad Dt**	**TM**	**TLC**	**Size/ Mean**	**Std Dev**	**NR/G PS**	**Off**	**Amn**	**Civ**	**Off**	**Amn**	**Civ**	**Off**	**Amn**	**Civ**
83	8	10/14/2004	3/10/2005			24	0										
84								AMD0		144							
85	9	12/14/2004	5/6/2005			24	0										
86								AMD0		144							
87	10	2/18/2005	7/6/2005			24	0										
88								AMD0		144							
89	11	4/19/2005	9/1/2005			24	0										
90								AMD0		144							

The first column (A) lists the block number. The block number is incremented for each new block of instruction. Column B lists the course content number (CC #). The course content number represents a distinct piece of the course content within the block and increases incrementally throughout the block. The user can continue to use the incremented series when a new block starts, or start the series of numbers over again with the new block. Column C is the washback point or rate—the probability that a person washes back at a designated point in the syllabus. The course content column (D) is informational only and relates the name of the course content to the course content number. Column E is also informational and indicates the training method. A single course content number can contain multiple training methods (e.g., lecture, demonstration, observation, or testing). We will also use the term *UOI* for these training methods. Column G lists the length of any training method in hours. The sum of hours for each training method (or each UOI) is the total length of the course content number.

Figure 3.32
Course Plans of Instruction Sheet (Screen 1 of 3)



L3ABR3P031 007

Security Forces Apprentice
Plans of Instruction

343 TRS

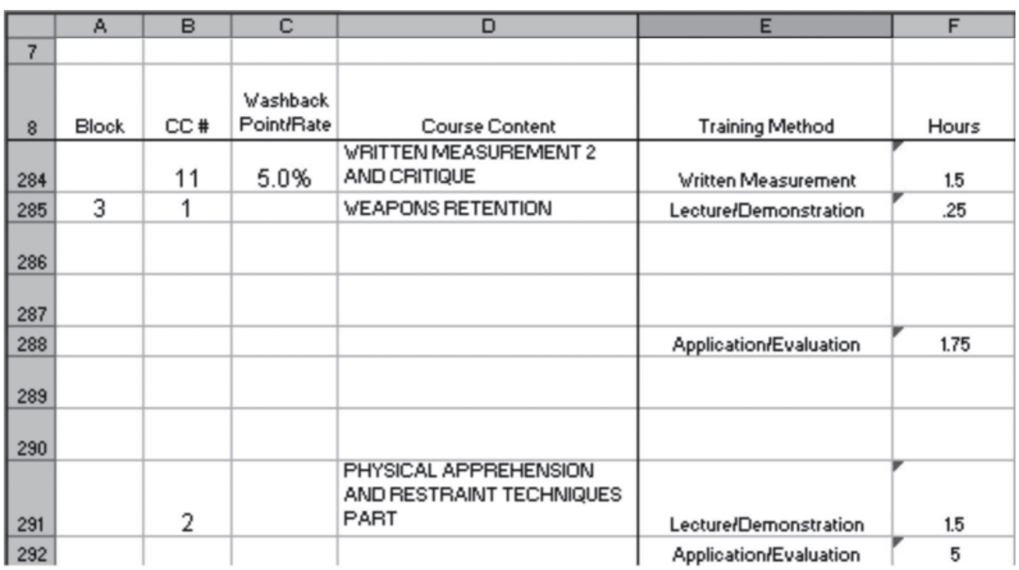
Devices

Return to Si

	A	B	C	D	E	F	G	H	I
8	Block	CC #	Washback Point/Rate	Course Content	Training Method	Hours	Duty Days (Hours)	Full Complete (Piece)	Instructors
9									
10	1	1		COURSE ORIENTATION	Lecture	1			2
11		2		CAREER FIELD HISTORY	Lecture	1			2
12									
13									
14									
15									
16		3		SECURITY FORCES CODE OF CONDUCT	Lecture	1.5			2

Figure 3.33 shows an example with a new block, 3 CC 1, starting after block 2 CC 11. Block 3 CC 1 has two training methods (lecture/demonstration and application/evaluation). The total length of CC 1, called "Weapons Retention," is two hours.

Figure 3.33
Course Content Sample with Multiple Training Methods
(Course Plans of Instruction Sheet)



	A	B	C	D	E	F
7						
8	Block	CC #	Washback Point/Rate	Course Content	Training Method	Hours
284		11	5.0%	WRITTEN MEASUREMENT 2 AND CRITIQUE	Written Measurement	1.5
285	3	1		WEAPONS RETENTION	Lecture/Demonstration	.25
286						
287						
288					Application/Evaluation	1.75
289						
290						
291		2		PHYSICAL APPREHENSION AND RESTRAINT TECHNIQUES PART	Lecture/Demonstration	1.5
292					Application/Evaluation	5

Referring again to Figure 3.32, columns G and H give the user the ability to extend the standard duty day to accommodate units that need to be completed in one sitting. Column G lists the length of the duty day and column H expresses whether or not the piece of instruction must be continuous. Normally, the length of the duty day is defined in the school policies sheet (see Figures 3.15 and 3.19). Column G gives the user the ability to override the length of the normal duty day for this particular piece of instruction. By putting a number in the column greater than the duty-day length, this piece of instruction will continue up to the specified length of time.

Column H can be used to represent a demonstration that must be completed the same day it is started and cannot be paused overnight or over a weekend. This feature is called *chunking*. A single training method or multiple training methods can be chunked together. The chunk cannot begin unless there is adequate time in the day to finish. If a chunk is longer than the standard duty day, the duty day must be extended. The user designates whether or not a piece of instruction needs to be chunked by entering a value into Column H, which is labeled "Full Complete (Piece)." If only one piece of instruction is chunked, the user enters a 1 in column H. If two pieces of instruction need to be completed during one sitting, the user enters a 1 in column H in the first row and a 2 in column H in the second row. The user can chunk as many pieces of instruction as can be completed during one sitting.

Finally, column I (also depicted in Figure 3.32) lists the number of instructors required to teach the particular training method of a particular course content unit and block. In Figure 3.32, block 1 CC 1, training method: lecture, requires two instructors for one hour; CC 2 requires two instructors for one hour and CC 3 requires two instructors for 1.5 hours.

Figure 3.34 displays the next third of the course plans of instruction sheet. For each training method, the model requires the user to list all the training devices (column J) and the quantity (column K) for that particular training method. If devices are needed for more than one training method, they must be repeated after each training method in which they are required. The training device name (column M) is informational and not necessary. The training device stock numbers in column J must match exactly the training device stock numbers in the training device sheet (see Figure 3.37).

At the end of each training method, the training devices are returned to storage. If they are part of a room, the user should not identify them as separate items. Rather, the user would define a specific facility (classroom) name that would include the special equipment in the room and make sure that the room is specified in the facility requirements portion of the page (see Figure 3.23).

Column L (see Figure 3.34) is a placeholder. The designed intent is to give the user the option of locking down the device or permitting the sharing of devices among training methods in a given class. In this way, the device is not returned to storage at the end of each training method but is kept "checked out" to the course across methods. This prevents other classes from using the device until the class releases it. A good example is the case of rebuilding an engine. Once the class starts working on an engine (the training device), it may be necessary for the class to completely finish with it before returning it for other classes to use. In this case,

Figure 3.34
Course Plans of Instruction Sheet (Screen 2 of 3)

	A	B	C	D	J	K	L	M
1	L3ABR3P031 007							
2					im Control			
3	Security Forces Apprentice							
4	Plans of Instruction							
5								
6								
7								
8	Block	CC #	Washback Point/Rate	Course Content	Training Device (Stock #)	Training Device (Qty)	Lock Device (Piece)	Training Device (Name)
9								
10	1	1		COURSE ORIENTATION				
11		2		CAREER FIELD HISTORY				
12					6730004026437	1		SCRN_PRJ_6730004026437
13					701001 COMPUTER SYSTEM	1		SYS_COM_PC_GENERIC_701001
14					7195008218995	1		PODIUM_7195008218995
15					PXGE850	1		PRJ_CD_SHARP_PXGE850
16		3		SECURITY FORCES CODE OF CONDUCT				
17					6730004026437	1		SCRN_PRJ_6730004026437
18					701001 COMPUTER SYSTEM	1		SYS_COM_PC_GENERIC_701001
19					7195008218995	1		PODIUM_7195008218995
20					PXGE850	1		PRJ_CD_SHARP_PXGE850

the user identifies the engine as a locked device (a lock value of 1). As long as the device has a 1 in the lock column, it will not be returned to storage. This prevents any other class from using or sharing the device during the period of the lock.

The last third of the course plans of instruction sheet (see Figure 3.35) describes the facility requirements for each training method. The model allows a maximum of three different types of facilities (columns N, P, and R) to be required at the same time. Each facility requirement is followed by a quantity column (O, Q, and S, respectively) representing the number of the facility types required. Facilities are treated much like training devices—they are "retrieved" when needed and "returned to stock" when no longer in use.

The course plans of instruction sheet captures all the resources and related information needed for training. Handling these critical data in this way is the primary reason why the model is data-driven and not code-driven. In the schoolhouse model, rather than having to make code changes for any change in the POI, the code interprets and follows the directions, which are given in the POI in a very generic manner, negating the need for multiple and different models for each AETC course.

The course plans of instruction sheet is closely paired with the course resource sheet. Both sheets must have the same course number.

Figure 3.35
Course Plans of Instruction Sheet (Screen 3 of 3)

	A	B	C	D	N	O	P	Q	R	S
1	L3ABR3P031 007									
2										
3	Security Forces Apprentice									
4	Plans of Instruction									
5										
6										
7										
8	Block	CC #	Washback Point/Rate	Course Content	Classroom/Facility Type	#	Classroom/Facility Type	#	Classroom/Facility Type	#
9										
10	1	1		COURSE ORIENTATION	Classroom	1				
11		2		CAREER FIELD HISTORY	Classroom	1				
12										
13										
14										
15										
16		3		SECURITY FORCES CODE OF CONDUCT	Classroom	1				

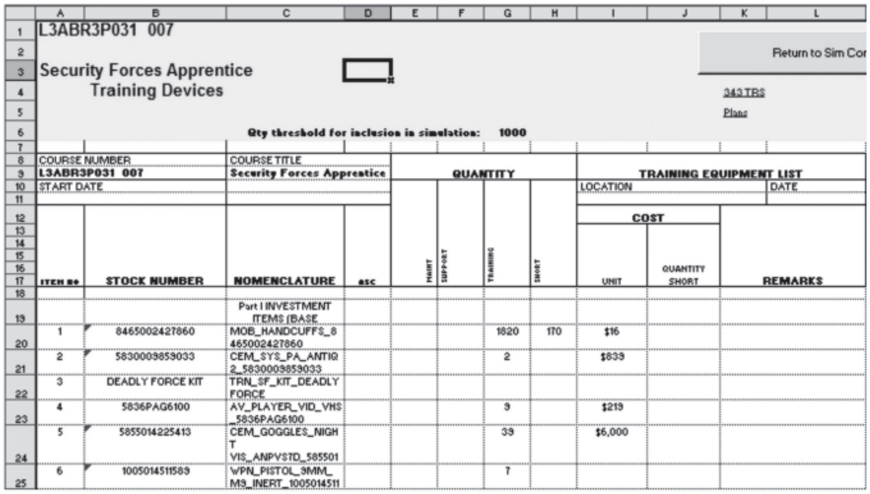
TRS Course Resource Sheet

Each course has its own course resource sheet. It uses the course number exactly as input (e.g., L3ABR3P031 007) with the word "Devices" appended. The course resource sheet is very similar in appearance to Air Force Form 120 (see Figure 3.36). The plans of instruction sheet references devices on the devices sheet. The stock number, column J on the plans of instruction sheet (Figure 3.34), must match a stock number in column B on the devices sheet (Figure 3.36). The nomenclature, column M on the plans of instruction sheet and column C on the devices sheet, does not need to match but will generate a warning if there is a mismatch. The plans of instruction sheet details when and how many devices are needed in the POI. The device sheet records the available inventory.

The total available inventory of devices is dependent upon selections made on an earlier sheet. On the policy setup sheet (see Figure 3.18), the user can select whether the inventory includes the training column (column G in Figure 3.36), the support column (column F in Figure 3.36), or both (Figure 3.18). At a minimum, the training column is usually selected. The model does not utilize information in columns E or H. Information for all the columns is output into a text file for use in postprocessors.

In Figure 3.36, there is a caption, "Qty threshold for inclusion in simulation," followed by a number. The number following the caption is the maximum number of any one training device that the simulation will use. The purpose of this number is to eliminate calculations using large numbers of consumables (e.g., ammunition). If the total number of devices exceeds the maximum quantity, the model will ignore the training device. Figure 3.36 shows a value of 1,000, but the user can change the maximum quantity to any appropriate value. As the value increases, there may be some corresponding increase in run time.

Figure 3.36
Training Device Sheet



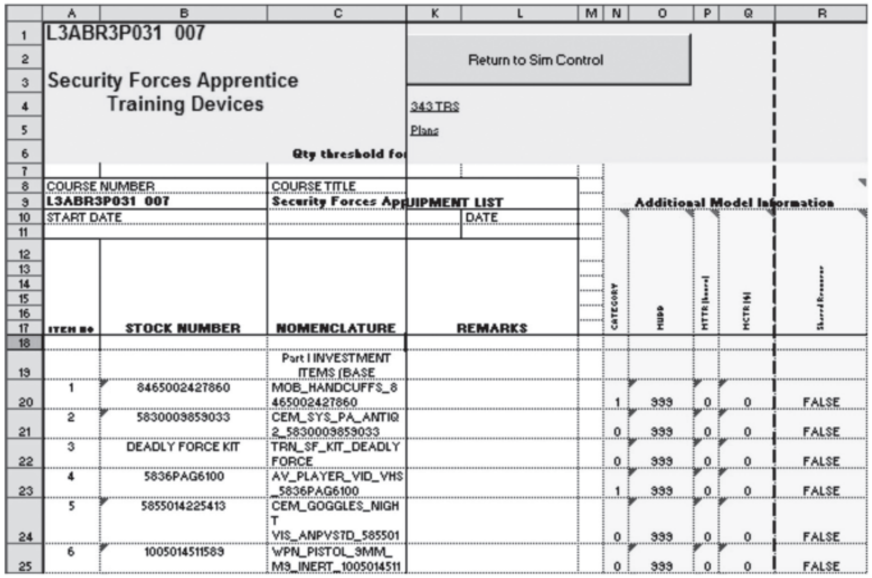
ITEM NO	STOCK NUMBER	NOMENCLATURE	ASC	MAINT	SUPPORT	TRAINING	SHORT	UNIT	QUANTITY SHORT	REMARKS
		Part I INVESTMENT ITEMS (BASE								
1	8465002427860	MOB_HANDCUFFS_8 465002427860				1820	170	$16		
2	5830009859033	CEM_SYS_PA_ANTIQ 2_5830009859033				2		$839		
3	DEADLY FORCE KIT	TRN_SF_KIT_DEADLY FORCE								
4	5836PAG6100	AV_PLAYER_VID_VHS _5836PAG6100				9		$219		
5	5855014225413	CEM_GOGGLES_NIGH T VIS_ANPVS7D_585501				39		$6,000		
6	1005014511589	WPN_PISTOL_9MM_ M9_INERT_1005014511				7				

On each devices sheet are five additional columns of data for simulation use (see Figure 3.37). Some of the data have been implemented in the model while others have not. The first column, N, contains category data. The effect of the category value is dependent on the model policy chosen (see Figures 3.15 and 3.16) on the policy sheet. There are four possible values for category. As described earlier, category 3 represents a critical device that cannot be worked around. These items will either generate a delay or a training deficiency, depending on the model policy chosen. Category 2 devices are important but not critical. They generate a work-around and are recorded in the output. For most policies, category 2 shortages will not affect the production of the schoolhouse. Categories 1 and 0 are not modeled in the simulation. They can be used for postprocess analysis as they are output into text files.

Columns O through R are reliability, maintainability, and cost placeholders. These columns are not currently used. Future model improvements may implement these data for more precise cost calculations. Reliability is specified in terms of mean uses between breaks (MUBB) and is calculated by dividing the mean time between failures (MTBF) by the average usage time. Unfortunately, no reliability data exist currently for training device breakdown. Column P, mean time to repair (MTTR), is expressed in hours. If this feature were implemented in the future, a broken device would not be available to the schoolhouse until this delay time has past, representing completion of device repairs. Column Q, mean cost to repair (MCTR), would be charged every time a device breaks.

Column R, shared resource, is the last placeholder on the page. It represents a future capability that would allow other courses to share the resource. Currently, the model uses only training device items requested in a course from its specific corresponding course device sheet.

Figure 3.37
Training Device Additional Information (Training Device Sheet)



Creating Text Input Files

At the end of the data input effort, the user returns to the simulation-control sheet of the model. The model is designed to store all the data in a scenario directory. In Figure 3.38, the user selects the first radio button, "Folder under this model directory" and then types in a folder name for the scenario. The final step is to click the first button below ("Prepare Specified Data Folder for Later Simulation Runs"). The model then prompts the user for a scenario name and the current model is saved and renamed according to the format specified in Figure 3.39. Excel then confirms the new name and location of the file (Figure 3.40).

Figure 3.38
Example of Creating a New Scenario Directory

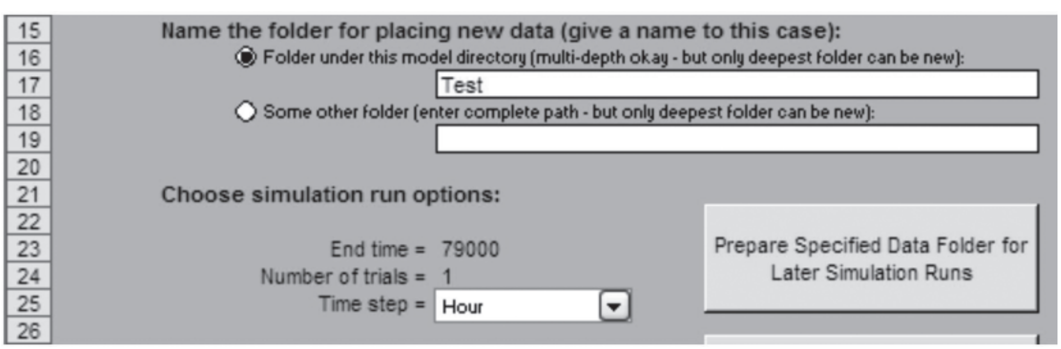


Figure 3.39
Example of Naming the Scenario Excel File

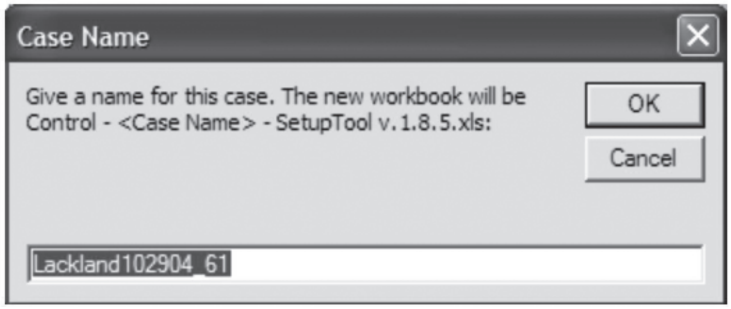


Figure 3.40
Example of Program Response to Scenario Naming

At this point, if the user wants to save the file, the "Save As" option must be used. At the same time that the model creates a copy of the Excel workbook as a new file, it also creates, in the same directory, numerous small text files that the Extend simulation requires. The text box on the next page displays a partial listing of the text files created.

Strengths and Weaknesses of the Excel Front End

With this detailed description of how to use the Excel front end, we now reflect briefly on the benefits and limitations of this approach. The front end has three major strengths. First, since Excel is a well-known user interface, most individuals are very comfortable using it to import data. Second, Excel's multiple pages allow the user to rigorously organize the various data elements and store the entire combined database for a schoolhouse in one workbook in one location. Third, due to the customization capabilities of Excel, the data pages have been designed to look like the actual Air Force data products.

Case Directory of Extend Files Created for Sample Case

341 TRS.txt

342 TRS.txt

. . .

345 TRS.txt

37 TRG Instructors.txt

ATTENTION-Frontend Notes.txt

ATTENTION-Input Notes.txt

Classrooms In.txt

Classrooms.txt

Control | Lackland102904_61 | SetupTool v.1.8.5.xls

Control.ini

Devices In.txt

Directory.txt

Dorms In.txt

Group-School-Course Names | Indexed from 1 in history.txt

History.txt

Instructors In.txt

L3ABP1C231 000 All Devices | Form 120 style.txt

L3ABP1C231 000 Device Type | Indexed from 0 in history.txt

L3ABP1C231 000 Plans.txt

L3ABR2S032 000 All Devices | Form 120 style.txt

L3ABR2S032 000 Device Type | Indexed from 0 in history.txt

L3ABR2S032 000 Plans.txt

. . .

L3AQR4D031 004 All Devices | Form 120 style.txt

L3AQR4D031 004 Device Type | Indexed from 0 in history.txt

L3AQR4D031 004 Plans.txt

SchForPost.txt

Students In.txt

NOTE: In the example, these files were placed in c:\Documents and Settings\Administrator\My Documents\Production\Schoolhousev.1.8.7\Test.

The weakness of our Excel front end is in one primary area: Currently, data must be entered into it by hand or through a variety of ad hoc processing steps. Thus, it is time-consuming and laborious to build the databases. Part of the reason is that AETC does not currently have a standardized data system for storing all of these data. In fact, the front end is a de facto integrated database. In the future, AETC intends to automate course information more fully, which will greatly reduce the time to build a schoolhouse model database. Currently, it can take between one day and one week to gather and input all the data required for a course. Since courses change on a fairly regular basis, a lot of data input is required to keep the databases up to date.

Once the front end has been used to create text files for the Extend model, the data input process is complete and the Excel case workbook can be closed. In the next chapter, we give a brief overview of the Extend simulation.

CHAPTER FOUR

The Extend Simulation

Introduction

The engine of the schoolhouse model is the simulation tool written in Extend. As we have already discussed, the purpose of the model is to assist in the determination of resources needed to accomplish the training mission. A variety of methodological approaches could have been selected for this purpose. We begin this chapter by comparing alternative analytical tools, discussing the strengths and weaknesses of the simulation approach. This leads naturally to a description of the measures of performance used in the model. We then turn to a brief structural overview of the simulation model.

Why Simulation?

Within the tool kit of prescriptive and descriptive modeling techniques, simulation plays a significant role in its ability to capture detail and complexity that other closed formed methods (e.g., steady-state or optimization models) cannot. Simulation models permit not only a more extensive representation of the structure and flows of a system, but also the collection of a broader range of performance measures and the testing of more comprehensive policies. However, this advantage is not without its costs, including the difficulty in developing a more complicated model, establishing its credibility, collecting more extensive input data, and managing significantly longer calculations and larger result databases. Additionally, although all models should be appropriately applied within an analytic framework specific to the problem under investigation, simulations can sometimes be particularly difficult to employ properly.

The primary reason we chose a simulation instead of an inventory, queuing, or other simpler model is because we desire to discover not only the primary constraints in the training process, but also the secondary and tertiary restrictions. One of our primary questions is, What does it take to increase the number of trainees in a specialty by a certain number? Among instructors, training devices, classrooms, or other facilities, AETC training managers are usually well aware of the primary resource constraints prohibiting greater throughput. Similarly, simpler models could also more easily identify the initial system bottlenecks. However, all too often, relieving these primary constraints allows only a fraction of the desired production to be achieved. In reality, constraints are faced, and many of them are very detailed or caused by complex interaction effects with multiple parts of the training system. For example, a desired increase in the number of trained security police means the increased utilization of facilities such as firing ranges and gyms that are shared with other courses. Increasing the utilization of these facilities by security police would almost certainly have an adverse effect on the availability of these resources to other specialties that share these facilities and, therefore, on the training production in these specialties. Without a comprehensive model to capture the individual course needs, interactions with other courses, and possible work-arounds for all critical training resources, the calculated resource requirements would almost certainly be wrong. We believe that the required accuracy and fidelity needed for these calculations justifies the need for a simulation.

At the same time, we have developed the simulation to be as simple as possible. To a large extent, the simulation is just a bookkeeping device. It assures that the appropriate mix of resources is available and used for each course, records their usage, and recycles when resource use has been completed. There are, of course, subtle complexities included in this system, particularly the various policies used in training. For example, the simulation allows blocks of training to be resequenced if required resources are not immediately available. These details are precisely what has required us to use simulation as the technique of choice.

To assist in the application of the simulation, we have developed several tools to reduce the effort needed to operate the simulation and analyze the results. In the previous chapter, we described the front end, which logically structures the input data and provides a convenient way of managing the input data. The model itself produces an extensive history file that records every event for postsimulation analysis. A sample SAS program, provided in Appendix B, parses the history file and can be used to gather statistics.

As with most modeling tools, the manner in which the model is used must fit the analytic purposes. One run of the simulation tells us little more than the number of trainees completing courses and the operational impact on the training system. To be effective for determining required resources and desirable policies, comparative simulations must be performed. For example, to determine the resources required to train an additional 2,000 airmen in a particular specialty, we first run the simulation at the nominal number of trainees and determine the quantity of resources, including time, required to produce the baseline number of airmen. We then increase the desired number to be trained by 2,000 and compare the two cases to determine what additional resources are needed and what system impacts occur. Because the simulation represents the training process at a significant level of detail, a wide variety of operational details can be examined.

Measures of Performance

An explanation of the proper use of the simulation is a natural transition into a discussion of the measures of the training system performance. The primary training measure is the throughput—how many airmen in each of a number of specialties can be trained in a certain amount of time. As equally important as this rather comprehensive measure of benefit is a whole variety of measures of cost. Principal among these is the utilization and utilization rates of resources: trainers, training devices, and facilities. These elements translate directly into economic costs of the training system. For example, an increased number of aircraft maintainers can be trained, but at the cost of a set of additional instructors, an additional training aircraft mock-up, and an additional classroom. The simulation identifies these needed resources in the history file so that a postprocessing cost model can assess the fiscal impact.

The simulation also provides secondary measures of throughput and costs. For example, not all airmen are able to complete training or to complete it the first time through. The simulation calculates those who "wash out" of the training process completely, or "wash back" from a particular training block and must repeat the instruction. Washouts reduce training throughput. Washbacks have a more complicated impact in that they compete for admittance in future classes with new trainees. Because washbacks may have to wait before rejoining a class, training administrators count this lost time as IIT and seek to reduce it. The simulation also captures this measure. Comparative analysis can determine the effect of additional resources, particularly class size and policy alternatives for reducing the IIT rate.

Limitations in resources could cause temporary or long-term impacts on the cost and quality of training. The simulation is able to devise a number of different types of work-arounds, such as resequencing blocks of instruction. These work-arounds preserve throughput, but at some level of increased effort (or at least increased administration cost) and with a potential reduction in training quality (increased washouts or washbacks, or decreased trainee proficiency). Additionally, work-arounds may not be possible. Training delays, which reduce throughput, or training deficiencies, which reduce training quality, may be the only solutions. The simulation is able to capture these measures as well.

The Simulation Model's Structure

Essentially, four different types of entities are modeled: classes of students, instructors, classrooms, and training devices. In Figure 4.1 (which does not depict all elements in detail), the four horizontal blocks represent the path of the four entity types. They meet in the vertical box titled "Class Meeting" with data collected in the vertical box titled "After-Class Follow-Up" before the entities return to the beginning of the pipeline structure.

Figure 4.2 is our overview of the schoolhouse process. The class (also called a flight) is a collection of students receiving the same instruction. It is the initiating entity and drives the movement of all other entities in the model, including instructors, training devices, and facilities.[1] The POI guides the flow of the class and the need for resources. Each class follows the POI based on the availability of the required instructional resources. The POI defines the content of instruction, the length of the instruction, the number of instructors, the number and type of facilities, and the number and type of training devices. If any of the resources are not available, the model is designed to shuffle the order of content

[1] Perhaps it is difficult to think of a classroom or facility as an object "moving" through the system to meet up with a class of students, a number of instructors, and a group of training devices. In reality, the classroom does not move, but since there is no transit time within the model, there is no implication that the resources are actually moving.

Figure 4.1
Overview of the Extend Simulation

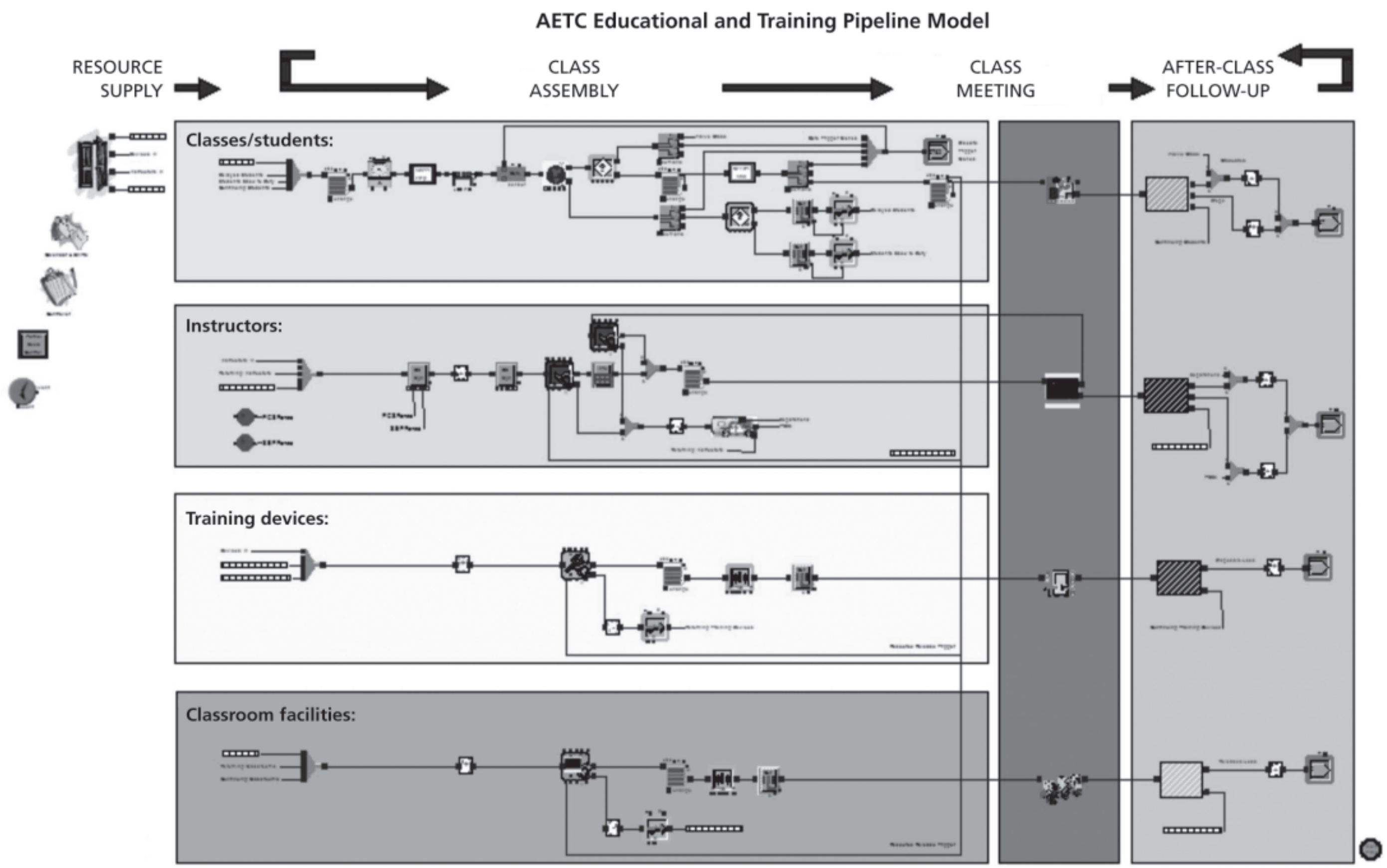
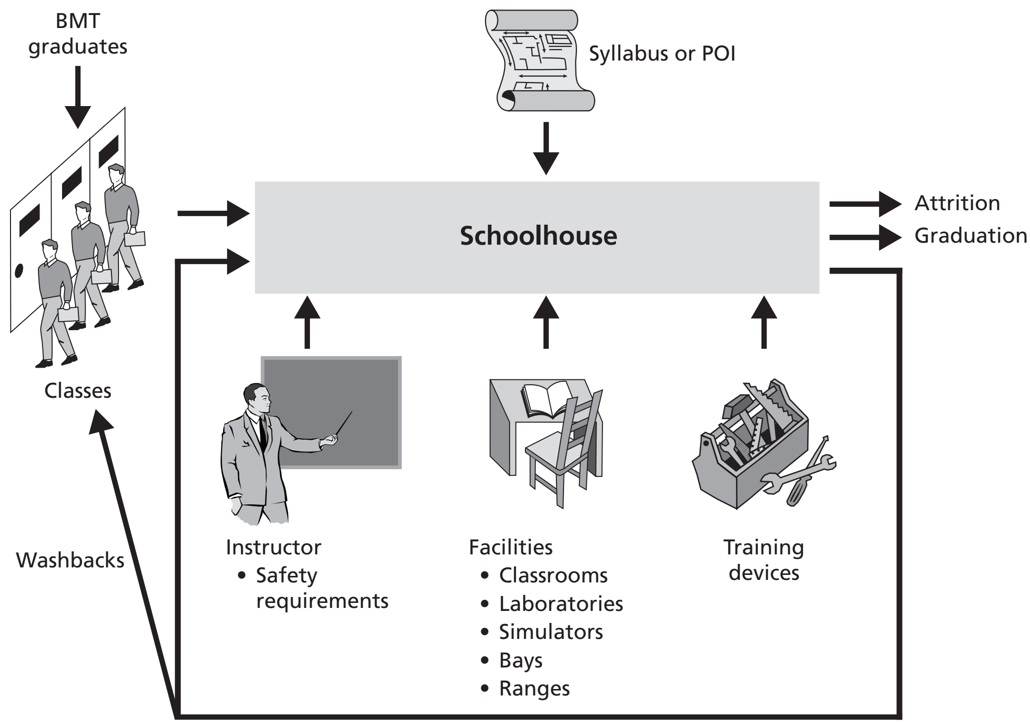


Figure 4.2
Overview of the RAND Schoolhouse Model



RAND *TR378-4.2*

(pieces of instruction) within a block, but at some point, a persistent resource shortage will delay a class or record a training deficiency (depending on the policy chosen). Consequently, the actual availability of each resource determines how well classes flow through their plans of instruction.

Flights represent the group of trainees in a class. Every class maintains a class roster. The roster is updated as individuals wash out of the class or wash back into other classes. The class maintains a record of where it is in the plan of instruction. The class also maintains other statistical information related to technical training.

The model is not designed to display the output data. Rather, the model produces a detailed record of every event (see Figures 4.4 through 4.7) that occurs over time into a history file. Postprocessors can then be used to scour the data and produce the required summary outputs and metrics.

Strengths and Weaknesses of the Extend Simulation

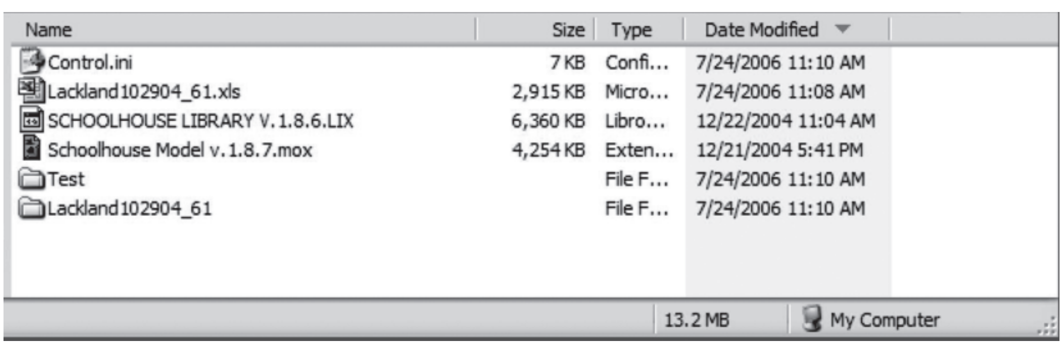
The Extend implementation of the schoolhouse has three main strengths over previous approaches. First, AETC has little capability to model the technical training process and so any repeatable mathematical tool is a marked increase over the present capability. Second, AETC manages hundreds of different courses. It is not feasible to build a model for every different course. The use of Extend facilitates a schoolhouse model that is data-driven, meaning that no new coding is required to model a different course. Finally, the model produces a detailed event history file of every event in the simulation. The model need not be rerun to look at other measures or metrics. By simply analyzing the event history file with statistical tools, new metrics and measures can be evaluated.

The weaknesses of the Extend simulation fall into two areas. The first major weakness is the long simulation run time. Processing the cases for changes at a typical wing can require run times of 10 to 20 computer hours. Dual-processor computers or multiple computers can reduce the time required. The second weakness was also listed above as a strength—the model records a detailed event history file that is extremely useful for analysis. Unfortunately, the file is very large. One two-hour run can easily produce a 100- to 200-MB file. Multiple replications require gigabytes of storage. The entire AETC pipeline may require terabytes of storage.

Running the Model

To start the Extend simulation, double-click the .mox file, such as "Schoolhouse Model v.1.8.7.mox" (see Figure 4.3). The "Control.ini" file points Extend to the directories and files needed for the case to be run.[2] When the model is ready, the screen should look like Figure 4.1. At this point, we are ready to run the simulation model. Figure 4.3 depicts a list of the required files. Assuming that we have just created the scenario directory "Test" or that "Test" was the last scenario directory created, the "Control.ini" file will be directly related to the scenario "Test." From the menu, choose Run | Run Simulation, or press Ctrl+R on the keyboard.

Figure 4.3
Required Extend Files



[2] The case that has just been set up by the Excel front end will be listed automatically in the "Control.ini" file. Assuming that we have just created the scenario "Test" or that "Test" was the last scenario created, the "Control.ini" file will point to that subdirectory and that case will be loaded automatically into the Extend simulation as it starts up.

The model creates one file during its execution (“History.txt”). Postprocessors use the data in “History.txt” to create summary measures of interest.

Understanding the Output

The history file records the key events of the simulation. In addition to its use as the source of model results, the “History.txt” file is very useful for debugging. Events are recorded in chronological order, making it easy to recreate and examine the processing of classes, the state of the resources, or the status of training at any time in the simulation. The simulation does not produce any metrics or measures of performance. Instead, postprocessors read in the history file and can provide a wide variety of summary measures describing the scenario and the resultant execution of the training plans.

The columns of data in the “History.txt” file (an example of which appears in Figure 4.8) vary in usage depending on the type of event (identified by an event code) recorded in the history file. Figures 4.4 through 4.7 depict the usage of the various columns of data for each event code. Figures 4.4 and 4.6 show events associated with classes and instructors (events 10 through 190). Figures 4.5 and 4.7 contain events associated with training devices and facilities (events 200 through 390).

Figure 4.8 depicts a small excerpt of a “History.txt” file. Using Figures 4.4 through 4.7, we can interpret the file. The first column lists the run number; the first run is numbered 0. This convention of naming the first item “0” saves space and some additional programming.[3]

The second column represents the time at which the event occurs. For example, the first several events occur at 226.5. This run is using hours, so 226.5 hours have elapsed. The third column contains the event code. In our example, in the first row, the event code is 20, representing “Flight ready for UOI part.” For event code 20, we will use the information in Figures 4.4 and 4.6 to interpret the columns (for event numbers higher than 190, we use Figures 4.5 and 4.7).

The fourth column is the entity classification (0 for a class of students, 1 for instructors, 2 for training devices, and 3 for facilities). The fifth column is the identification number of the particular entity. Here we have class identification number 2. The sixth column is the type of entity. Since there is only one type of class and only one type of instructor, for those events associated with classes or instructors, the type value will always be zero.

The seventh and eight columns represent the school (also called the squadron) and course. In this case, the entire excerpt concerns squadron number 3, course number 1.[4] The ninth,

---

[3] The software uses matrices with a first column that lists zero. We could ignore the first location in a matrix, but that wastes space. We could just subtract one from our visible value, but that adds some extra minor programming. Care should be exercised with this convention. If at any point we forget to add or subtract one when translating a value, we have introduced an error into the system.

[4] This is different from the numbering convention noted previously. Here, the number “1” is used for the first input. If we were to cross-reference to our input data, the third squadron is the 343rd TRS and the first course is the only course, the Security Forces Apprentice course.

Figure 4.4
Events 10 Through 190 (Columns 1 Through 11)

Run	Time	Event Code	(Event Label - not outputted to history file)	Entity	Parameters...					
1	2	3		4	5	6	7	8	9	10	11
Run	Time	0	Flight initialized (in-place for warm start only)	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	10	New flight enters system (as accessions)	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	15	Student from SITS into flight roster	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	20	Flight ready for UOI part	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	21	Flight skips to later CC	0	ID	Type	School	Course	Block	OrigCC	NewCC
Run	Time	22	Policy 1 class proceeds w/ unavailable resources	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	23	Policy 2 class proceeds w/ Cat2 workaround(s)	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	24	Policy 2 class delayed for instructor/room/Cat3 device(s)	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	25	Policy 3 class proceeds w/ Cat2 workaround(s)	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	26	Policy 3 class deficiency from instructor/room/Cat3 device(s)	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	27	Policy 4 class deficiency from Cat2 or instructor/room/Cat3 de	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	28	Policy 5 class delayed for Cat2 or instructor/room/Cat3 device	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	30	Class(es) delayed - full-complete reqt exceeds remaining dut	0	ID	Type	School	Course	Block	CC	FirstUOI
Run	Time	35	Flight diverts to study hall	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	40	Flight moves to assemble	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	50	Flight proceeds to UOI part	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	60	Flight completes UOI part	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	65	Student washes back - leaves flight roster into SITS	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	70	Flight interrupted after UOI part	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time										
Run	Time	89	Student drops out and departs simulation	0	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	90	Flight graduates and departs simulation	0	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	100	Instructor initialized (in-place for warm start only)	1	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	110	Instructor enters sytem	1	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	120	Instructor ready for UOI part	1	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	130	Instructor interrupted before assembly	1	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	140									
Run	Time	150	Instructor proceeds to UOI part	1	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	160	Instructor completes UOI part	1	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	170	Instructor interrupted after UOI part	1	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	180									
Run	Time	189	Instructor PCSs and departs simulation	1	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	190	Instructor separates and departs simulation	1	ID	Type	School	Course	<blank>	<blank>	<blank>

Figure 4.5
Events 200 Through 390 (Columns 1 Through 11)

Run	Time	Event Code	(Event Label - not outputted to history file)	Entity	Parameters...						
1	2	3		4	5	6	7	8	9	10	11
Run	Time	200	Device initialized (in-place for warm start only)	2	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	210	Device enters system	2	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	220	Device ready for UOI part	2	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	230	Device interrupted before assembly	2	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	240									
Run	Time	250	Device proceeds to UOI part	2	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	260	Device completes UOI part	2	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	270	Device interrupted after UOI part	2	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	280									
Run	Time	290	Device disposed and departs simulation	2	ID	Type	School	Course	<blank>	<blank>	<blank>
Run	Time	300	Classroom initialized (in-place for warm start only)	3	ID	Type	School	<blank>	<blank>	<blank>	<blank>
Run	Time	310	Classroom enters system	3	ID	Type	School	<blank>	<blank>	<blank>	<blank>
Run	Time	320	Classroom ready for UOI part	3	ID	Type	School	<blank>	<blank>	<blank>	<blank>
Run	Time	330	Classroom interrupted before assembly	3	ID	Type	School	<blank>	<blank>	<blank>	<blank>
Run	Time	340									
Run	Time	350	Classroom proceeds to UOI part	3	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	360	Classroom completes UOI part	3	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	370	Classroom interrupted after UOI part	3	ID	Type	School	Course	Block	CC	UOI Part
Run	Time	380									
Run	Time	390	Classroom removed and departs simulation	3	ID	Type	School	<blank>	<blank>	<blank>	<blank>

Figure 4.6
Events 10 Through 190 (Columns 12 Through 19+)

Event Implemented in simulator
Field not yet reported by simulator

Run	Time	Event Code	(Event Label - not outputted to history file)									
1	2	3		12	13	14	15	16	17	18	19	...
Run	Time	0	Flight initialized (in-place for warm start only)	TOS								
Run	Time	10	New flight enters system (as accessions)	TOS	StartStudentID	EndStudentID						
Run	Time	15	Student from SITS into flight roster	StudentIn	StudentOut							
Run	Time	20	Flight ready for UOI part	Time Spent								
Run	Time	21	Flight skips to later CC									
Run	Time	22	Policy 1 class proceeds w/ unavailable resources	Resource	Category	Available	Reqt	Resource	Category	Available	Reqt	...
Run	Time	23	Policy 2 class proceeds w/ Cat2 workaround(s)	Resource	Category	Available	Reqt	Resource	Category	Available	Reqt	...
Run	Time	24	Policy 2 class delayed for instructor/room/Cat3 device(s)	Resource	Category	Available	Reqt	Resource	Category	Available	Reqt	...
Run	Time	25	Policy 3 class proceeds w/ Cat2 workaround(s)	Resource	Category	Available	Reqt	Resource	Category	Available	Reqt	...
Run	Time	26	Policy 3 class deficiency from instructor/room/Cat3 device(s)	Resource	Category	Available	Reqt	Resource	Category	Available	Reqt	...
Run	Time	27	Policy 4 class deficiency from Cat2 or instructor/room/Cat3 de	Resource	Category	Available	Reqt	Resource	Category	Available	Reqt	...
Run	Time	28	Policy 5 class delayed for Cat2 or instructor/room/Cat3 device	Resource	Category	Available	Reqt	Resource	Category	Available	Reqt	...
Run	Time	30	Class(es) delayed - full-complete reqt exceeds remaining dut	LastUOI								
Run	Time	35	Flight diverts to study hall									
Run	Time	40	Flight moves to assemble									
Run	Time	50	Flight proceeds to UOI part									
Run	Time	60	Flight completes UOI part	Time Spent								
Run	Time	65	Student washes back - leaves flight roster into SITS	StudentIn								
Run	Time	70	Flight interrupted after UOI part	InterruptCode								
Run	Time											
Run	Time	89	Student drops out and departs simulation	StudentID	TOS							
Run	Time	90	Flight graduates and departs simulation	TOS								
Run	Time	100	Instructor initialized (in-place for warm start only)	TOS	Grade	PCS	SEP					
Run	Time	110	Instructor enters sytem	TOS	Grade	PCS	SEP					
Run	Time	120	Instructor ready for UOI part	TOS	Grade	PCS	SEP	Time Spent				
Run	Time	130	Instructor interrupted before assembly	TOS	Grade	PCS	SEP					
Run	Time	140										
Run	Time	150	Instructor proceeds to UOI part	TOS	Grade	PCS	SEP					
Run	Time	160	Instructor completes UOI part	TOS	Grade	PCS	SEP	Time Spent				
Run	Time	170	Instructor interrupted after UOI part	TOS	Grade	PCS	SEP	InterruptCode				
Run	Time	180										
Run	Time	189	Instructor PCSs and departs simulation	TOS	Grade	PCS	SEP					
Run	Time	190	Instructor separates and departs simulation	TOS	Grade	PCS	SEP					

Figure 4.7
Events 200 Through 390 (Columns 12 Through 19+)

Event Implemented in simulator
Field not yet reported by simulator

Run	Time	Event Code	(Event Label - not outputted to history file)									
1	2	3		12	13	14	15	16	17	18	19	...
Run	Time	200	Device initialized (in-place for warm start only)	InServiceTime	LastRepair	NextBreak						
Run	Time	210	Device enters system	InServiceTime	LastRepair	NextBreak						
Run	Time	220	Device ready for UOI part	InServiceTime	LastRepair	NextBreak	Time Spent					
Run	Time	230	Device interrupted before assembly	InServiceTime	LastRepair	NextBreak						
Run	Time	240										
Run	Time	250	Device proceeds to UOI part	InServiceTime	LastRepair	NextBreak						
Run	Time	260	Device completes UOI part	InServiceTime	LastRepair	NextBreak	Time Spent					
Run	Time	270	Device interrupted after UOI part	InServiceTime	LastRepair	NextBreak	InterruptCode					
Run	Time	280										
Run	Time	290	Device disposed and departs simulation	InServiceTime	LastRepair	NextBreak						
Run	Time	300	Classroom initialized (in-place for warm start only)									
Run	Time	310	Classroom enters system									
Run	Time	320	Classroom ready for UOI part	Time Spent								
Run	Time	330	Classroom interrupted before assembly									
Run	Time	340										
Run	Time	350	Classroom proceeds to UOI part									
Run	Time	360	Classroom completes UOI part	Time Spent								
Run	Time	370	Classroom interrupted after UOI part	InterruptCode								
Run	Time	380										
Run	Time	390	Classroom removed and departs simulation	InServiceTime								

Figure 4.8
Excerpt of "History.txt" File

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	0	226.5	20	0	2	0	3	1	1	10	1					
2	0	226.5	40	0	2	0	3	1	1	10	1					
3	0	226.5	50	0	2	0	3	1	1	10	1					
4	0	226.5	150	1	129	0	3	1	1	10	1	0	0	9999999	9999999	
5	0	226.5	150	1	130	0	3	1	1	10	1	0	0	9999999	9999999	
6	0	226.5	250	2	966	20	3	1	1	10	1	0	9999999	9999999		
7	0	226.5	250	2	1012	40	3	1	1	10	1	0	9999999	9999999		
8	0	226.5	250	2	1234	44	3	1	1	10	1	0	9999999	9999999		
9	0	226.5	250	2	1257	47	3	1	1	10	1	0	9999999	9999999		
10	0	226.5	350	3	57	11	3	1	1	10	1					
11	0	227.5	60	0	1	0	3	1	1	12	3	5				
12	0	227.5	160	1	111	0	3	1	1	12	3	0	0	9999999	9999999	5
13	0	227.5	120	1	111	0	3	1				0	0	9999999	9999999	

tenth, and eleventh columns of the “History.txt” file (see Figure 4.8) list the block number, course content number, and unit of instruction part. Here, we see the class is at block 1 CC 10 and UOI part 1 of the syllabus. The Security Forces Apprentice course uses only one block.

CC 10 is “Military Law.” It is two hours in length and requires two instructors. The two hours of elapsed time will not show until an elapsed time of 228.5 as in event 60.[5] The two instructors appear as event 150 in the fourth and fifth rows of Figure 4.8. The following four rows (6 through 9) in Figure 4.8, event 250, show four training devices proceeding to assembly. The event in row 10, number 350, represents a facility also moving to assembly.[6]

At the end of the simulation, a message announces the completion of the simulation and the time elapsed. At this point, the user can close the Extend simulation and begin postprocessing the results. The analysis of the history file can be done in a number of ways. Appendix B contains a copy of a sample SAS program to tally key utilization measures for instructors, training devices, and facilities. Since each case is contained in a separate directory, postprocessing tools can also be used to determine changes in effectiveness measures across cases. The analytic process of understanding and synthesizing the event histories should not be understated. The purpose of the schoolhouse model is to examine a wide range of resource-oriented questions with regard to training. The Excel front end, the Extend simulation, and the postprocessing tools facilitate this analysis.

---

[5] The authors verified that 228.5 does show in the “History.txt” file. The event 60 at 229.5 is another class completing the same instruction.

[6] As noted previously, the training devices and facilities do not actually “move,” since the time to move is zero, but the idea of transiting or moving is a computer convention invoked to ensure that the items are captured by the classes, thereby preventing other classes from using the same facilities or devices at the same time.

CHAPTER FIVE

Next Steps

The schoolhouse model is a first step forward in modeling the technical training pipeline.[1] It gives the Air Force the ability to look at various impacts of resource limitations or training policies. The model is working and available now.

Currently, there are two primary users of the model, AETC SAS and RAND Project AIR FORCE (PAF). The Navy is investigating the use of the model for the Naval training pipeline. AETC SAS is using the model to make quick-turn order-of-magnitude estimates of program objective memorandum (POM) costs. PAF is using the model to study policy impacts and resource requirements in specific training pipelines.

There are a number of enhancements that can further improve the model. A next step in its development would be the creation of a model users group. The users group could prioritize and coordinate changes to the model so that there is no duplication of effort or development of incompatible variations of the model. In addition, a users group could approve philosophical approaches to model upgrades. There is more than one way to model training pipeline processes. It is important that a users group oversee the philosophical approaches so that the model remains valid. Table 5.1 provides a list of potential future enhancements to the model. It is not exhaustive, but rather a starting point for a users group to begin prioritizing model development.

Finally, there are a number of other approaches to analyzing the training pipeline and process that need to be fully explored. They may provide reasonable answers without the extensive run-time and data storage requirements of the subject simulation model.

---

1 There are a number of alternative methodologies that are not based on simulations and that could be used to analyze the training pipeline and process. The other approaches would likely provide reasonable answers without the extensive run-time and data storage requirements of the simulation model discussed in this user's manual. If run time and data storage should be issues for the Air Force, then it may choose to explore other methodologies. However, trends in computing power and cost make it likely that resource requirements necessary to run this particular model will decrease over time.

Table 5.1
Potential Model Improvements

Model Improvement	Comment
Multiple shifts	Currently, multiple shifts are modeled using separate runs and then added together.
Automated data input	Model input is tedious and time-consuming. Much of the data already exist in standardized databases and only requires software to translate the data into the schoolhouse model format.
Approved postprocessors	A standardized set of postprocessors with community-accepted and standardized business rules would help preclude contradicting results. A costing postprocessor would be especially useful.
Reporting options	Completion of the reporting options is a potential improvement (see Figure 3.10).
Personnel options	Completion of the detailed personnel options is a potential improvement (see Figure 3.20).
Instructor enhancements	An ability to differentiate instructors by grade, certification, and other measures is a potential improvement.
Training device maintenance	The completion of training device maintenance capabilities would also require the development of training device maintenance data.
Sharing training devices	Currently, training devices are shared only in one course among multiple classes. This enhancement would allow multiple courses and multiple classes to share the same device.

APPENDIX A

Technical Requirements for the Schoolhouse Model

Computer hardware must facilitate a combination of the Excel and Extend requirements.

Microsoft® Windows® requirements:

- Intel® Pentium® 233 MHz processor or higher
- 128 MB RAM or higher
- 200 MB of available hard-disk space for installation
- 10 GB or more of available hard-disk space for model runs
- Super VGA (800x600) or higher monitor

Macintosh® requirements:[1]

- Power Macintosh® G4 or later
- 128 MB RAM or higher
- operating system 9.1+ (Mac OS X® recommended)
- 200 MB of available hard-disk space for installation
- 10 GB or more of available hard-disk space for model runs
- Internet Explorer® 5 or later, or Netscape Communicator 5 or later

The Extend player version is available for free download from Imagine That, Inc., at http://www.imaginethatinc.com/support_downloads.html (as of July 12, 2006).

[1] The model has not been tested on a Macintosh, so we can only estimate the hardware requirements.

Sample SAS Code to Analyze "History.txt" File

This appendix contains sample SAS code for parsing and summarizing a "History.txt" file. Comments are listed throughout the code. The authors can be contacted to provide a machine-readable version of this code.

```
/*Macro reading in the different History.txt files*/
%let fnum=_np;
data TRAINING.Flight&fnum TRAINING.Instructor&fnum TRAINING.Device&fnum
TRAINING.Device_Def&fnum
TRAINING.Classroom&fnum;
Run=Field1; Time=Field2; Event_Code=Field3; Entity=Field4; Id=Field5; Type=Field6;
School=Field7;
Course=Field8; Block=Field9; POI=Field10; UOI=Field11;
    drop Field1 Field2 Field3 Field4 Field5 Field6 Field7 Field8 Field9 Field10 Field11;
        set TRAINING.TRAINING;

/*Separates the History.txt into four different data sets (flight, instructor, devices, class-
rooms) based on event codes*/
    if entity = 0
        then output TRAINING.Flight&fnum;
    else if event_code = 150 | event_code = 160
        then output TRAINING.Instructor&fnum;
    else if event_code = 250 | event_code = 260
        then output TRAINING.Device&fnum;
    else if event_code = 230 | event_code = 270
        then output TRAINING.Device_Def&fnum;
    else if event_code = 350 | event_code = 360
        then output TRAINING.Classroom&fnum;
    run;

/*Sorts the instructor data set*/
proc sort data =TRAINING.Instructor&fnum out=TRAINING.Inst_Start_Stop_Code&
fnum;
    by school course id time event_code;
```

```
run;

/*Adds the number of instructors based on the 150 event codes*/
data TRAINING.Inst_Sorted_Code&fnum;
    set TRAINING.Inst_Start_Stop_Code&fnum;
        if event_code = 150
            then instruct_number + 1;
        else if event_code = 160
            then instruct_number = instruct_number - 1;
run;

/*Calculates the difference in times for each instructor (instruction time)*/
%let Instructor_Hours = TRAINING.Inst_Sorted_Code&fnum;
data TRAINING.Inst_Calc_Manhrs&fnum;
    set &Instructor_Hours(keep=time event_code id school course instruct_number);

    retain tempTime 0;
    retain tempInstructor 0;

            duration = time - tempTime;
            manhours = tempInstructor * duration;
            tempTime = time;
            tempInstructor = instruct_number;
run;

/*Calculates the time for each instructor*/
data TRAINING.Inst_Leaving_Manhrs&fnum;
    set TRAINING.Inst_Calc_Manhrs&fnum;
        if event_code = 160 then do;
            _col_ = manhours;
            output;
        end;
run;

/*Sums the instructor manhours*/
%LET MAF = %NRSTR ((161.12));

PROC SQL;

PROC SQL;
CREATE TABLE TRAINING.Inst_Num_Needed&fnum AS SELECT DISTINCT inst_
leaving_manhrs.school,
    inst_leaving_manhrs.course,
    (SUM(inst_leaving_manhrs.manhours)) AS sum_of_manhours,
```

```
      (SUM(inst_leaving_manhrs.manhours)/&MAF) AS Inst_Reqrd
FROM TRAINING.inst_leaving_manhrs&fnum AS inst_leaving_manhrs
GROUP BY inst_leaving_manhrs.school, inst_leaving_manhrs.course ; QUIT;

%LET _EGTASKLABEL =;

run;
quit;

ODS _ALL_ CLOSE;
ODS LISTING;

proc means data=TRAINING.Inst_Num_Needed&fnum;
            var sum_of_manhours Inst_Reqrd;
            output out=TRAINING.inst_total_summary&fnum sum=sum_of_manhours
Inst_Reqrd;
run;
```

/*Final instructor data set*/

```
data TRAINING.Inst_Fnl&fnum;
      set TRAINING.Inst_Num_Needed&fnum TRAINING.inst_total_summary&fnum
            (keep=sum_of_manhours Inst_Reqrd);
run;

proc print data=TRAINING.Inst_Fnl&fnum;
run;
```

/*Sorts the devices data set*/

```
proc sort data=TRAINING.Device&fnum out=TRAINING.Device_Start_Stop_Code&
fnum;
            by id type school course time event_code;
run;
```

/*Adds the number of devices used*/

```
data TRAINING.Device_Sorted_Code&fnum;
      set TRAINING.Device_Start_Stop_Code&fnum;

            if event_code = 250
                  then number_of_devices + 1;
            if event_code = 250
                  then devices_number + 1;
            if event_code = 260
                  then devices_number = devices_number - 1;
run;
```

/*Calculates the hours each device is used*/

```
%let Devices_Hours = TRAINING.Device_Sorted_Code&fnum;
data TRAINING.Devices_Manhours&fnum;
    set &Devices_Hours(keep=time event_code id type school
        course number_of_devices devices_number);
        retain tempTime 0;
        retain tempdevices 0;

        duration = time - tempTime;
        devices_hrs = tempdevices * duration;
        tempTime = time;
        tempdevices = devices_number;
run;

data TRAINING.Device_Leaving_Manhrs&fnum;
set TRAINING.Devices_Manhours&fnum;
        if tempdevices = 0 then do;
            _col_ = devices_hrs;
            output;
        end;

run;
```

/*Sums the device hours*/

```
%LET _EGTASKLABEL = %NRBQUOTE(Query1);
PROC SQL;

PROC SQL;
CREATE TABLE TRAINING.Devices_Used&fnum AS SELECT Device_Leaving_
Manhrs.school,
    Device_Leaving_Manhrs.course, Device_Leaving_Manhrs.type, Device_Leaving_
Manhrs.id,
    (SUM(Device_Leaving_Manhrs.devices_hrs)) AS sum_hrs
FROM TRAINING.Device_Leaving_Manhrs&fnum AS Device_Leaving_Manhrs
GROUP BY Device_Leaving_Manhrs.school, Device_Leaving_Manhrs.course, Device_
Leaving_Manhrs.type, Device_Leaving_Manhrs.id; QUIT;

%LET _EGTASKLABEL =;

run;
quit;

ODS _ALL_ CLOSE;
ODS LISTING;
```

```
PROC SQL; DROP VIEW TRAINING.Devices_Tot_Per_Crs&fnum;

PROC SQL;
CREATE TABLE TRAINING.Devices_Tot_Per_Crs&fnum AS SELECT Devices_Used.
School,
     Devices_Used.Course,
     Devices_Used.Type,
     Devices_Used.Id,
     Devices_Used.sum_hrs,
     (SUM(Devices_Used.sum_hrs)) AS Sum_Hrs_Per_Crs
FROM TRAINING.Devices_Used&fnum AS Devices_Used
GROUP BY Devices_Used.School, Devices_Used.Course ; QUIT;

%LET _EGTASKLABEL =;

run;
quit;

ODS _ALL_ CLOSE;
ODS LISTING;

proc means data=TRAINING.Devices_Tot_Per_Crs&fnum;
          var sum_hrs;
          output out=TRAINING.Tot_Dev_Used_Sum&fnum sum=sum_hrs;
run;

/*Final devices data set*/
data TRAINING.device_fnl&fnum;
     set   TRAINING.Devices_Tot_Per_Crs&fnum   TRAINING.Tot_Dev_Used_Sum&
fnum (keep=sum_hrs);
run;

proc print data=TRAINING.device_fnl&fnum;
run;

/*Sorts the classroom data set*/
proc sort data = TRAINING.Classroom&fnum out=TRAINING.Sorted_Class_Code&
fnum;
          by school course id type time event_code;
run;
```

/*Adds the number of classrooms used*/

```
data TRAINING.Sorted_by_Class_Code&fnum;
	set TRAINING.Sorted_Class_Code&fnum;

		if event_code = 350
			then classroom_number + 1;
		if event_code = 360
			then classroom_number = classroom_number - 1;
```

/*Calculates the time each classroom is used*/

```
%let Classroom_Hours = TRAINING.Sorted_by_Class_Code&fnum;
data TRAINING.Classroom_Manhours&fnum;
	set &Classroom_Hours(keep=time event_code id type school course classroom_
number);

			retain tempTime 0;
			retain tempclassroom 0;
			duration = time - tempTime;
			classroom_hrs = tempclassroom * duration;
			tempTime = time;
			tempclassroom = classroom_number;
run;

data TRAINING.Class_Leaving_Hrs&fnum;
	set TRAINING.Classroom_Manhours&fnum;
		if tempclassroom = 0 then do;
			_col_ = classroom_hrs;
		output;
		end;

run;
```

/*Sums the classroom hours*/

```
%LET _EGTASKLABEL = %NRBQUOTE(Query1);
PROC SQL;

PROC SQL;
CREATE TABLE TRAINING.Classrooms_Used&fnum AS SELECT Class_Leaving_Hrs.
school,
	Class_Leaving_Hrs.course, Class_Leaving_Hrs.type, Class_Leaving_Hrs.id,
	(SUM(Class_Leaving_Hrs.Classroom_hrs)) AS sum_hrs
FROM TRAINING.Class_Leaving_Hrs&fnum AS Class_Leaving_Hrs
GROUP BY Class_Leaving_Hrs.school, Class_Leaving_Hrs.course, Class_Leaving_Hrs.
type, Class_Leaving_Hrs.id; QUIT;
```

```
%LET _EGTASKLABEL =;

run;
quit;

ODS _ALL_ CLOSE;
ODS LISTING;

%LET _EGTASKLABEL = %NRBQUOTE(Query1 for class_tot_final_summary_lt);
PROC SQL;

PROC SQL;
CREATE TABLE TRAINING.Class_Hr_Per_Crs&fnum AS SELECT Classrooms_Used.
School,
      Classrooms_Used.Course,
      Classrooms_Used.Type,
      Classrooms_Used.Id,
      Classrooms_Used.sum_hrs,
      (SUM(Classrooms_Used.sum_hrs)) AS Sum_Hr_Per_Crs
FROM TRAINING.Classrooms_Used&fnum AS Classrooms_Used
GROUP BY Classrooms_Used.School, Classrooms_Used.Course ; QUIT;

%LET _EGTASKLABEL =;

run;
quit;

ODS _ALL_ CLOSE;
ODS LISTING;

proc means data=TRAINING.Class_Hr_Per_Crs&fnum;
var sum_hrs;
output out=TRAINING.Class_Total_Summary&fnum sum=sum_hrs;
run;

/*Final classroom data set*/
data TRAINING.Class_Fnl&fnum;
set TRAINING.Class_Hr_Per_Crs&fnum TRAINING.Class_Total_Summary&fnum
(keep=sum_hrs);
run;

proc print data=TRAINING.Classroom_Manhours&fnum;
            var time event_code classroom_number;
            sum classroom_hrs;
run;
```

APPENDIX C

Data Definitions

Table C.1 lists descriptions of the data addressed in each sheet in the model, along with their respective variable types and additional comments on each element. The last column in the table lists the figure(s) that illustrates these elements in Chapter Three of this user's guide.

Table C.1
Data Descriptions

Sheet	Description	Variable Type	Comment	Figure Reference
Sim Control sheet	Use model versions available in the following Master Directory	Character	Must be in a directory format	3.4
Sim Control sheet	Name the folder for placing new data	Character	No special characters in name	3.4
Policy Setup sheet	Additional non–school days/holidays	mm/dd/yy	Maximum of 199 entry lines	3.19
TRG Setup sheet	Courses	Character	Must match exactly every reference	3.21, 3.22
TRG Setup sheet	Facility available	Integer		3.21, 3.23
TRG Setup sheet	Group-managed meeting facility	Character	Describes the group-managed facility	3.24
TRG Setup sheet	Group-managed meeting facility: Hourly cost	Real	Not required	3.24
TRG Setup sheet	Group-managed meeting facility: Available	Integer	Not required	3.24
TRG Setup sheet	Other group facility	Character	Describes the other group facilities	3.25
TRG Setup sheet	Other group facility: Hourly cost	Real	Not required	3.25
TRG Setup sheet	Other group facility: Assignment cost (one-time)	Real	Not required	3.25
TRG Setup sheet	Other group facility: Available	Integer	Not required	3.25
TRG Setup sheet	Other costs	Character	Not required	3.25
TRG Setup sheet	Other costs: Yearly cost	Real	Not required	3.25
TRG Setup sheet	Other costs: Hourly cost	Real	Not required	3.25
TRG Setup sheet	Other costs: One-time cost	Real	Not required	3.25
Squadron sheet	Course Number	Character	Must match exactly	3.28
Squadron sheet	Course Title	Character	Not required	3.28
Squadron sheet	# Days	Integer	Not required	3.28
Squadron sheet	PDS	Character	Not required	3.28

Table C.1—Continued

Sheet	Description	Variable Type	Comment	Figure Reference
Squadron sheet	TM	Character	Not required	3.28
Squadron sheet	PM	Character	Not required	3.28
Squadron sheet	Groups		Not required	3.29
Squadron sheet	Flight Mean Size		Not required	3.29
Squadron sheet	Flight Size Std Dev		Not required	3.29
Squadron sheet	Min Flight		Not required	3.29
Squadron sheet	Max Flight		Not required	3.29
Squadron sheet	Total Entries		Not required	3.29
Squadron sheet	Rooms Added		Not required	3.29
Squadron sheet	Avg Washback		Not required	3.29
Squadron sheet	Program Elimination	Real		3.29
Squadron sheet	Current Elimination	Real		3.29
Squadron sheet	Repeat Intervals (hours)		Not required	3.29
Squadron sheet	Certified instructors: GS11	Integer		3.30
Squadron sheet	Certified instructors: GS12	Integer		3.30
Squadron sheet	Certified instructors: E3	Integer		3.30
Squadron sheet	Certified instructors: E4	Integer		3.30
Squadron sheet	Certified instructors: E5	Integer		3.30
Squadron sheet	Certified instructors: E6	Integer		3.30
Squadron sheet	Certified instructors: E7	Integer		3.30
Squadron sheet	Course Schedules: Course	Character	Must match exactly	3.31

Table C.1—Continued

Sheet	Description	Variable Type	Comment	Figure Reference
Squadron sheet	Course Schedules: FY	Integer	Not required	3.31
Squadron sheet	Course Schedules: Sequence	Character	Not required	3.31
Squadron sheet	Course Schedules: PM	Character	Not required	3.31
Squadron sheet	Course Schedules: CL/NR	Integer	Not required	3.31
Squadron sheet	Course Schedules: Start Dt	mm/dd/yy		3.31
Squadron sheet	Course Schedules: Grad Dt	mm/dd/yy	Not required	3.31
Squadron sheet	Course Schedules: TM	Character	Not required	3.31
Squadron sheet	Course Schedules: TLC	Character	Not required	3.31
Squadron sheet	Course Schedules: Size/Mean	Real	Can be integer form	3.31
Squadron sheet	Course Schedules: Std Dev	Real	Can be integer form	3.31
Squadron sheet	Course Schedules: NR/GPS	Character	Not required	3.31
Squadron sheet	Course Schedules: Off	Integer	Not required	3.31
Squadron sheet	Course Schedules: Amn	Integer	Not required	3.31
Squadron sheet	Course Schedules: Civ	Integer	Not required	3.31
POI sheet	Block	Integer		3.32
POI sheet	CC #	Integer		3.32
POI sheet	Washback Point/Rate	Real		3.32
POI sheet	Course Content	Character		3.32
POI sheet	Training Method	Character		3.32, 3.33
POI sheet	Hours	Real or integer		3.32
POI sheet	Duty Days (Hours)	Integer		3.32

Table C.1—Continued

Sheet	Description	Variable Type	Comment	Figure Reference
POI sheet	Full Complete (Piece)	Integer		3.32
POI sheet	Instructors	Integer		3.32
POI sheet	Training Device (Stock #)	Character	Must match exactly a device on the resource sheet	3.34
POI sheet	Training Device (Qty)	Integer	Exceptions: 1ea, 2ea, or any whole number plus “ea”	3.34
POI sheet	Lock Device (Piece)	Integer		3.34
POI sheet	Training Device (Name)	Character		3.34
POI sheet	Classroom/Facility Type	Character	Must match a facility on TRG setup sheet	3.35
POI sheet	#	Integer		3.35
POI sheet	Classroom/Facility Type	Character	Must match a facility on TRG setup sheet	3.35
POI sheet	#	Integer		3.35
POI sheet	Classroom/Facility Type	Character	Must match a facility on TRG setup sheet	3.35
POI sheet	#	Integer		3.35
Training Device sheet	Qty threshold for inclusion in simulation	Integer		3.36
Training Device sheet	ITEM NO	Integer	Not required	3.36
Training Device sheet	STOCK NUMBER	Character		3.36
Training Device sheet	NOMENCLATURE	Character	Not required	3.36
Training Device sheet	ASC	Character	Not required	3.36
Training Device sheet	QUANTITY: MAINT	Integer	Not required	3.36
Training Device sheet	QUANTITY: SUPPORT	Integer		3.36
Training Device sheet	QUANTITY: TRAINING	Integer		3.36

Table C.1—Continued

Sheet	Description	Variable Type	Comment	Figure Reference
Training Device sheet	QUANTITY: SHORT	Integer	Not required	3.36
Training Device sheet	COST: UNIT	Real	Not required	3.36
Training Device sheet	COST: QUANTITY SHORT	Integer	Not required	3.36
Training Device sheet	REMARKS	Character	Not required	3.36
Training Device sheet	Additional Model Information: CATEGORY	Integer	0, 1, 2, or 3 only	3.37
Training Device sheet	Additional Model Information: MUBB	Real	Not required (future implementation)	3.37
Training Device sheet	Additional Model Information: MTTR (hours)	Real	Not required (future implementation)	3.37
Training Device sheet	Additional Model Information: MTCR ($)	Real	Not required (future implementation)	3.37
Training Device sheet	Additional Model Information: Shared Resource	Character	Not required (future implementation)	3.37

Bibliography

Banks, Jerry, and John S. Carson, *Discrete-Event Simulation*, Englewood Cliffs, N.J.: Prentice Hall, 1984.

Der, Geoff, and Brian S. Everitt, *A Handbook of Statistical Analyses Using SAS*, 2nd ed., Boca Raton, Fla.: CRC Press, 2001.

Imagine That, Inc., homepage for Extend product information and downloads, undated. Online at http://www.imaginethatinc.com/ (as of March 24, 2006).

Law, Averill M., and W. David Kelton, *Simulation Modeling and Analysis*, 2nd ed., New York: McGraw-Hill, 1991.

Manacapilli, Thomas, Bart Bennett, Lionel A. Galway, and Joshua Weed, *Air Education and Training Command Cost and Capacity System: Implications for Organizational and Data Flow Changes*, Santa Monica, Calif.: RAND Corporation, MR-1797-AF, 2004. Online at http://www.rand.org/pubs/monograph_reports/MR1797/ (as of July 12, 2006).

Microsoft Corporation, Microsoft Office Online Excel product page, undated. Online at http://office.microsoft.com/en-us/FX010858001033.aspx (as of March 24, 2006).